



**TECNOLÓGICO  
NACIONAL DE MÉXICO**



## **Programación Orientada a Objetos – Licenciatura en Informática**

### **Tema 2. Clases y Objetos**

#### **Integrantes:**

- **Contreras Herrera Ángel de Jesús – 21010180**
- **Kevin Alan Ortiz Flores – 21010207**

**Grupo: 2ª3B**

**Fecha de Entrega: 27 de marzo del 2023**



## Introducción.

La orientación a objetos es un paradigma de programación que facilita la creación de software de calidad por sus factores que potencian el mantenimiento, la extensión y la reutilización del software generado bajo este paradigma.

La programación orientada a objetos trata de amoldarse al modo de pensar del hombre y no al de la máquina. Esto es posible gracias a la forma racional con la que se manejan las abstracciones que representan las entidades del dominio del problema, y a propiedades como la jerarquía o el encapsulamiento.

El elemento básico de este paradigma no es la función (elemento básico de la programación estructurada), sino un ente denominado objeto. Un objeto es la representación de un concepto para un programa, y contiene toda la información necesaria para abstraer dicho concepto: los datos que describen su estado y las operaciones que pueden modificar dicho estado, y determinan las capacidades del objeto.

Java incorpora el uso de la orientación a objetos como uno de los pilares básicos de su lenguaje.

Podemos definir objeto como el "encapsulamiento de un conjunto de operaciones (métodos) que pueden ser invocados externamente, y de un estado que recuerda el efecto de los servicios". [Piattini et al., 1996].

Un objeto además de un estado interno, presenta una interfaz para poder interactuar con el exterior. Es por esto por lo que se dice que en la programación orientada a objetos "se unen datos y procesos", y no como en su predecesora, la programación estructurada, en la que estaban separados en forma de variables y funciones.

Las clases son abstracciones que representan a un conjunto de objetos con un comportamiento e interfaz común.

Podemos definir una clase como "un conjunto de cosas (físicas o abstractas) que tienen el mismo comportamiento y características... Es la implementación de un



tipo de objeto (considerando los objetos como instancias de las clases)". [Piattini et al., 1996].

Una clase no es más que una plantilla para la creación de objetos. Cuando se crea un objeto (instanciación) se ha de especificar de qué clase es el objeto instanciado, para que el compilador comprenda las características del objeto.

En este reporte abarcaremos la creación, uso, aplicación y reutilización de los objetos, así como sus respectivas reglas a emplear a la hora de crearlas, a su vez veremos unos ejemplos de estos, como se componen, como se han creado en base a la lógica y petición del problema empleado



## Competencias Específicas.

Específica(s):

Comprende y aplica la estructura de clases para la creación de objetos y utiliza clases predefinidas para facilitar el desarrollo de aplicaciones

Genéricas:

- Solución de problemas
- Capacidad de aplicar los conocimientos en la práctica
- Capacidad de aprender
- Capacidad de generar nuevas ideas (creatividad)
- Habilidad para trabajar en forma autónoma.

## Marco Teórico

Aunque parezca una obviedad, la base de la Programación Orientada a Objetos es el objeto. En la vida real todos los objetos tienen una serie de características y un comportamiento. Por ejemplo, una puerta tiene color, forma, dimensiones,



material... (goza de una serie de características) y puede abrirse, cerrarse... (posee un comportamiento). En Programación Orientada a Objetos, un objeto es una combinación de unos datos específicos y de las rutinas que pueden operar con esos datos. De forma que los dos tipos de componentes de un objeto son:

- Campos o atributos: componentes de un objeto que almacenan datos. También se les denomina variables miembros. Estos datos pueden ser de tipo primitivo (boolean, int, double, char...) o, a su vez, de otro tipo de objeto (lo que se denomina agregación o composición de objetos). La idea es que un atributo representa una propiedad determinada de un objeto.
- Rutinas o métodos: es una componente de un objeto que lleva a cabo una determinada acción o tarea con los atributos. En principio, todas las variables y rutinas de un programa de Java deben pertenecer a una clase. De hecho, en Java no hay noción de programa principal y las subrutinas no existen como unidades modulares independientes, sino que forman siempre parte de alguna clase.

## **Clase.**

Una clase representa al conjunto de objetos que comparten una estructura y un comportamiento comunes. Una clase es una combinación específica de atributos y métodos y puede considerarse un tipo de dato de cualquier tipo no primitivo. Así, una clase es una especie de plantilla o prototipo de objetos: define los atributos que componen ese tipo de objetos y los métodos que pueden emplearse para trabajar con esos objetos. Aunque, por otro lado, una clase también puede estar compuesta por métodos estáticos que no necesitan de objetos (como las clases construidas en los capítulos anteriores que contienen un método estático main).

## **Declaración de clase.**

Una vez que el paquete se ha declarado, la clase se puede definir después de eventuales líneas de importación.



Si intenta declarar dos clases de tipo public en un mismo archivo fuente, IntelliJ IDEA reacciona como sigue:

Una clase se declara con la palabra clave class, seguida por el nombre que ha elegido. Como para el paquete, este nombre debe empezar por una letra o por un guion bajo (\_). A continuación, puede contener letras, cifras y guiones bajos. Evite utilizar caracteres acentuados y opte por el formato CamelCase. Según las convenciones de nomenclatura más extendidas

### **Miembros de una Clase.**

Los miembros de una clase son un conjunto de elementos que definen a los objetos (atributos ó propiedades), así como los comportamientos o funciones (métodos) que maneja el objeto.

Entonces tenemos que una clase es la estructura de un objeto, es decir, la definición de todos los elementos de que está hecho un objeto.

Las clases ya tienen miembros que representan sus datos y comportamiento. Los miembros de una clase incluyen todos los miembros declarados en la clase, junto con todos los miembros declarados en todas las clases de su jerarquía de herencia.

- CAMPO: Los campos son variables declaradas en el ámbito de clase. Un campo puede ser un tipo numérico integrado o una instancia de otra clase. Por ejemplo, una clase de calendario puede tener un campo con la fecha actual.
- -CONSTANTES: Las constantes son campos o propiedades cuyo valor se establece en tiempo de compilación y no se puede cambiar.
- PROPIEDADES: Las propiedades son métodos de una clase a los que se obtiene acceso como si fueran campos de esa clase. Una propiedad puede proporcionar protección a un campo de clase con el fin de evitar que se cambie sin el conocimiento del objeto.



- **METODOS:** Los métodos definen las acciones que una clase puede realizar. Los métodos pueden aceptar parámetros que proporcionan datos de entrada y devolver datos de salida a través de parámetros. Los métodos también pueden devolver un valor directamente, sin usar ningún parámetro.
- **EVENTOS:** Los eventos proporcionan a otros objetos notificaciones sobre lo que ocurre, como clics en botones o la realización correcta de un método. Los eventos se definen y desencadenan mediante delegados.
- **OPERADORES:** Los operadores sobrecargados se consideran miembros de clase. Si se sobrecarga un operador, se define como método estático público en una clase. Los operadores predefinidos (+, \*, <, etc.) no se consideran miembros.
- **INDIZADORES:** Los indizadores permiten indizar un objeto de manera similar a como se hace con las matrices.
- **CONSTRUCTORES:** Los constructores son métodos a los que se llama cuando el objeto se crea por primera vez. Se usan a menudo para inicializar los datos de un objeto.
- **DESTRUCTORES:** En C#, los destructores se usan en raras ocasiones. Son métodos a los que llama el motor de ejecución del runtime cuando el objeto está a punto de quitarse de la memoria. Generalmente se utilizan para asegurarse de que los recursos que se deben liberar se controlan apropiadamente.
- **TIPOS ANIDADOS:** Los tipos anidados son tipos declarados dentro de otro tipo. Los tipos anidados se usan a menudo para describir objetos utilizados únicamente por los tipos que los contienen.

#### **AMBITO REFERENTE A UNA CLASE.**

El ámbito es el contexto que tiene un nombre dentro de un programa, determina en qué partes del programa una entidad puede ser usada.



Esto sirve para que se pueda volver a definir una variable con un mismo nombre en diferentes partes del programa sin que haya conflictos entre ellos.

Si una variable es declarada dentro de un bloque (método/función/procedimiento), ésta será válida solo dentro de ese bloque y se destruirá al terminar el bloque. Adicionalmente, la variable no podrá verse ni usarse fuera del bloque (en el exterior del bloque). La variable dentro del bloque es una variable local y solo tiene alcance dentro del bloque que se creó y sus bloques hijos, pero no en bloques hermanos ni padres, una variable definida fuera de cualquier bloque es una variable global y cualquier bloque puede acceder a ella y modificarla.

En el caso de programación orientada a objetos (POO), una variable global dentro de una clase es llamada variable de instancia, y cada objeto creado con esa clase tiene una. Adicionalmente existen variables globales que son comunes a un todo el objeto creado con una clase y son llamadas variables de clase.

Hay dos tipos de alcances, el estático que también es llamado lexicográfico, donde el alcance se determina en tiempo de compilación, mientras que las variables de alcance dinámico se verificasen en el hilo de ejecución.

## **Especificadores de Acceso**

Un especificador (modificador) de acceso o alcance es una palabra clave (reservada) que nos permite controlar nivel de alcance que tiene cada variable, método o clase. Existen tres public, protected, private, static, final.

public establece un nivel de acceso total, es decir una clase public puede ser usada por otras clases del mismo paquete y de otros paquetes no relacionados, lo mismo para los métodos y variables.

public es el especificador de acceso más utilizado en estudiantes ya que no nos enfrentamos a la dificultad (¿..?) de tener que determinar que se puede y que no se puede utilizar en una clase.





`private` establece un nivel restringido de acceso, En el cual los miembros declarados como tal solo son accesibles dentro de la clase, aunque la clase herede de esta última. las variable y métodos `private` son usados cuando los clientes de una clase no deben preocuparse (ni les debe importar) como se efectúan las acciones claves de un programa. establecer miembros `private` establece el mayor nivel de seguridad.

`protected` establece un nivel de acceso intermedio entre `public` y `private` en el que los miembros solo son accesibles por clases y métodos dentro del mismo paquete.

`static` el modificador `static` crea variables y métodos de clase (a esto es lo que nos referimos cuando hablamos de variables de clase) estos objetos pueden ser llamados desde cualquier parte de la clase modificando sus valores reales de forma definitiva. la diferencia con una variable normal es que cuando llamamos a una variable de instancia (normal) cambiamos su valor solo en esa instancia (métodos, etc..) en cambio una variable estática cambia de valor definitivamente cada vez que se llama, en palabras simples actualizamos su valor real desde cualquier parte del programa.

`final` el modificador `final`, crea una constante, en realidad en java no existen las constantes, pero `final` hace que las variables imiten su comportamiento. Cualquier intento por modificar una variable `final` creara un error en tiempo de ejecución.

Un ejemplo de `final`, es la variable `PI`, de la clase `Math`

### **Creación de Objetos.**

Nota: una clase puede tener muchos objetos diferentes.



Una vez que se define una clase, el nombre de la clase se convierte en nuevo tipo de dato y se utiliza tanto como declarar una variable de ese tipo, como para crear objeto del mismo. La sintaxis para declarar un objeto es:

```
NombreClase nombreObjeto;
```

Ejemplo

```
Circulo miCirculo; //declara la variable mi circulo
```

La variable `miCirculo` es una instancia de la clase `Circulo`. La creación de un objeto de una clase se llama creación de una instancia de la clase. Un objeto es similar a una variable que tiene un tipo de clase. La creación de variables de un tipo de dato primitivo se realiza simplemente declarándolas, esta operación crea la variable y le asigna espacio en memoria.

#### 1.- Creación de la clase.

Una variable de tipo clase es una variable referencia, que puede contener la dirección de en memoria (o referencia) de un objeto de la clase o null para una referencia no valida. La declaración de un objeto simplemente asocia el objeto con una clase, haciendo al objeto una instancia de esa clase. La declaración no crear el objeto. Para crear realmente `miCirculo` (objeto de la clase `Circulo`) se necesita utilizar el operador **new** con el objeto de indicar a la computadora que cree un objeto `miCirculo` y asigne espacio de memoria para ella.

La sintaxis para crear un objeto es:

```
nombreObjeto = new NombreClase();
```

Ejemplo

La siguiente sentencia crea un objeto, `miCirculo`, y le asigna memoria:

```
miCirculo = new Circulo();
```



## Declaración e Instanciación

Se puede combinar la declaración y la instanciación en una sola sentencia con la siguiente sintaxis:

```
NombreClase nombreObjeto = new NombreClase();
```

### Ejemplo

Creación e instanciación de miCirculo en una etapa:

```
Circulo miCirculo = new Circulo();
```

## CLASES PREDEFINIDAS.

Uno de los puntos fuertes de Java es la gran cantidad de clases predefinidas que posee, dichas clases pueden ser utilizadas por los programadores sin necesidad de reinventar la rueda.

## CLASES CONTENEDORAS O WRAPPERS

En Java existen una serie de clases predefinidas equivalentes a los tipos primitivos denominadas wrappers, clases contenedoras o envoltorios, el identificador de cada una de estas clases es el mismo que el del tipo primitivo correspondiente, pero con la letra inicial en mayúsculas (salvo int - Integer y char - Character). Cada una de estas clases declaran un conjunto de métodos de gran utilidad; el uso de estas clases puede ser especialmente interesante para realizar determinadas operaciones mediante los métodos que implementan.

### - OBJETOS DE LA CLASE CHARACTER

La clase predefinida Character permite trabajar con instancias a las que se les puede asociar un único carácter Unicode. Esta clase incluye un conjunto de métodos que facilitan la manipulación de datos de tipo primitivo char.



El siguiente código muestra cómo crear referencias e instancias de la clase Character:

```
Character a1;  
a1 = new Character('A');  
Character a2 = new Character('B');
```

-

## OBJETOS DE LA CLASE STRING

String es una clase predefinida y especial de Java definida en la librería o paquete java.lang y orientada a manejar cadenas constantes de caracteres. Una instancia de la clase String es inmutable, es decir, una vez que se ha creado y se le ha asignado un valor, éste no puede modificarse (añadiendo, eliminando o cambiando caracteres). El siguiente código muestra diferentes ejemplos de creación de referencias e instancias de la clase String:

```
String cortesia = new String("Buenos días");  
// O bien, al ser una clase muy habitual en la forma abreviada:  
String saludo = "Hola";  
// O también:  
String despedida;  
despedida = "Adios";
```

## Definición, creación y reutilización de paquetes/librerías

Los paquetes Java agrupan las clases en librerías (bibliotecas). Los paquetes Java se utilizan de forma similar a como se utilizan las librerías en C++, sólo que en Java se agrupan clases y/o interfaces.

En los paquetes las clases son únicas, comparadas con las de otros paquetes, y permiten controlar el acceso; esto es, los paquetes proporcionan una forma de ocultar clases, evitando que otros programas o paquetes accedan a clases que son de uso exclusivo de una aplicación determinada.



Los paquetes se declaran utilizando la palabra reservada *package* seguida del nombre del paquete. Esta sentencia debe estar al comienzo del fichero fuente, concretamente debe ser la primera sentencia ejecutable del código Java, excluyendo, los comentarios y espacios en blanco. Por ejemplo:

```
package figuras;  
public class Circulo {  
    ...  
}
```

En este caso, el nombre del paquete es *figuras*. La clase *Circulo* se considera como parte del paquete. La inclusión de nuevas clases en el paquete es simple, se ha de colocar la misma sentencia al comienzo de los ficheros que contengan la declaración de las clases. Cada uno de los ficheros que contengan clases pertenecientes a un mismo paquete, deben incluir la misma sentencia *package*, y solamente puede haber una sentencia *package* por fichero. La sentencia *package* colocada el comienzo de un fichero fuente afectará a todas las clases que se declaren en ese fichero, se pueden referenciar paquetes precediendo con su nombre la clase que se quiere usar. También se puede usar la palabra reservada *import*, si se van a colocar múltiples referencias a un mismo paquete.

La sentencia *import* se utiliza para incluir una lista de paquetes en los que buscar una clase determinada, y su sintaxis es:

```
import nombre_paquete.Nombre_Clase;
```

Esta sentencia, o grupo de ellas, deben aparecer antes de cualquier declaración de clase en el código fuente.

## MANEJO DE EXCEPCIONES.



Una [excepción](#) es la indicación de que se produjo un error en el programa. Las excepciones, como su nombre lo indica, se producen cuando la ejecución de un [método](#) no termina correctamente, sino que termina de manera excepcional como consecuencia de una situación no esperada.

Cuando se produce una situación anormal durante la ejecución de un programa (por ejemplo se accede a un [objeto](#) que no ha sido inicializado o tratamos de acceder a una posición inválida en un [vector](#)), si no manejamos de manera adecuada el error que se produce, el programa va a terminar abruptamente su ejecución. Decimos que el programa deja de funcionar y es muy probable que el usuario que lo estaba utilizando ni siquiera sepa qué fue lo que pasó.

Cuando durante la ejecución de un [método](#) el computador detecta un error, crea un [objeto](#) de una [clase](#) especial para representarlo (de la [clase](#) Exception en Java), el cual incluye toda la información del problema, tal como el punto del programa donde se produjo, la causa del error, etc. Luego, "dispara" o "lanza" dicho [objeto](#) (throw en inglés), con la esperanza de que alguien lo atrape y decida como recuperarse del error. Si nadie lo atrapa, el programa termina, y en la consola de ejecución aparecerá toda la información contenida en el [objeto](#) que representaba el error. Este [objeto](#) se conoce como una [excepción](#). En el ejemplo 1 se ilustra esta idea.

### Material y Equipo:

- Computadora Portatil
- Eclipse Java IDE 2023-02 (Software)
- JavaSE-18 (JRE)
- Microsoft Word Office 2016

### Desarrollo de Practica.

Ejercicio1.

+ Persona
- String NomPers
- long TelPers



```
- byte EdadPers
- String NacionPers
- Fecha FechaPers
- char GenePers

+ Persona
+ Persona(String NomPers,long TelPers,byte EdadPers,String
NacionPers,Fecha FechaPers,char GenePers)
+ String getNomPers()
+ void setNomPers(String nomPers)
+ long getTelPers()
+ void setTelPers(long telPers)
+ byte getEdadPers()
+ void setEdadPers(byte edadPers)
+ String getNacionPers()
+ void setNacionPers(String nacionPers)
+ Fecha getFechaPers()
+ void setFechaPers(Fecha fechaPers)
+ char getGenePers()
+ void setGenePers(char genePers)
+ String toString()
```

```
+ Fecha
- byte dia
- byte mes
- short año

+ Fecha ()
+ Fecha (byte dia,byte mes,short año)
+ byte getDia()
+ void setDia(byte dia)
+ byte getMes()
+ void setMes(byte mes)
+ short getAño()
+ void setAño(short año)
+ String toString ()
```

Ejercicio 2.

```
+ Autor
- String Nombre
- String Apellido
```



```
+ Autor ( ) {}  
+ Autor(String Nombre,String Apellido)  
+ String getNombre()  
+ void setNombre(String nombre)  
+ String getApellido()  
+ void setApellido(String apellido)  
+ String toString()
```

+ Books

```
- String Titulo  
- Autor Autor  
- int ISBN  
- short NumPag
```

```
+ Books ( ) {}  
+ Books (String Titulo,Autor Autor,int ISBN,short NumPag)  
+ String getTitulo()  
+ void setTitulo(String titulo)  
+ Autor getAutor()  
+ void setAutor(Autor autor)  
+ int getISBN()  
+ void setISBN(int iISBN)  
+ short getNumPag()  
+ void setNumPag(short numPag)  
+ String toString()
```

Ejercicio3.

+ rentaAutos

```
- byte Dias  
- double KM  
- char Tipo
```

```
+ rentaAutos() {}  
+ rentaAutos(double KM,char Tipo, byte Dias)  
+ int getDias()  
+ void setDias(byte dias)  
+ double getKM  
+ void setKM(double kM)  
+ char getTipo  
+ void setTipo(char tipo)  
+ double detTarifa ()  
+ double CalCosto ()  
+ double CalMonto ()
```





#### Ejercicio 4.

+ Cuadrado
- float Lado
+ Cuadrado() {} + Cuadrado(float Lado) + float getLado() + void setLado(float lado) + float Area () + float Perimetro ()

+ Rectangulo
- float Altura - float Base
+ Rectangulo() + Rectangulo(float Altura,float Base) + float getAltura() + void setAltura(float altura) + float getBase() + void setBase(float base) + float Perimetro () + float Area()

+ Triangulo
- float Base - float Altura
+ Triangulo () + Triangulo (float Base,float Altura) + float getBase() + void setBase(float base) + float getAltura() + void setAltura(float altura) + float Hipotenusa3() + String TipoTriangulo + float Perimetro() + float Area() {

+ Circulo
- float radio
+ Circulo() + Circulo(float radio)



```
+ float getRadio()  
+ void setRadio(float radio)  
+ float Perimetro()  
+ float Area()
```

#### + Persona2

```
- String Nombre  
- byte Edad  
- char Sexo  
- float Peso  
- float Altura
```

```
+ Persona2()  
+ Persona2(String Nombre,byte Edad,char Sexo,float Peso,float Altura)  
+ Persona2(String Nombre,byte Edad,char Sexo)  
+ String getNombre()  
+ void setNombre(String nombre)  
+ byte getEdad()  
+ void setEdad(byte edad)  
+ char getSexo()  
+ void setSexo(char sexo)  
+ float getPeso()  
+ void setPeso(float peso)  
+ float getAltura()  
+ void setAltura(float altura)  
+ byte calcularMC()  
+ boolean esMayorDeEdad()  
+ String toString()  
+ String ToString()
```

## Resultados

Ejercicio 1

Persona.



```
eclipse-workspace - TemaClasesObjetos/src/TDA/Persona.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Cuadrado.java Rectangulo.java Triangulo.java Circulo.java TestFigurasGeometricas.java Persona2.java Persona.java
1 package TDA;
2
3 public class Persona {
4
5     private String NomPers;
6     private long TelPers;
7     private byte EdadPers;
8     private String NacionPers;
9     private Fecha FechaPers;
10    private char GenePers;
11
12    public Persona () {}
13
14    public Persona(String NomPers,long TelPers,byte EdadPers,String NacionPers,Fecha FechaPers,char
15
16        this.NomPers=NomPers;
17        this.TelPers=TelPers;
18        this.EdadPers=EdadPers;
19        this.NacionPers=NacionPers;
20        this.FechaPers=FechaPers;
21        this.GenePers=GenePers;
22    }
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631

```



```
eclipse-workspace - TemaClasesObjetos/src/TDA/Persona.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Cuadrado.java Rectangulo.java Triangulo.java Circulo.java TestFigurasGeometricas.java Persona2.java Persona.java
44 public void setEdadPers(byte edadPers) {
45     EdadPers = edadPers;
46 }
47
48 public String getNacionPers() {
49     return NacionPers;
50 }
51
52 public void setNacionPers(String nacionPers) {
53     NacionPers = nacionPers;
54 }
55
56 public Fecha getFechaPers() {
57     return FechaPers;
58 }
59
60 public void setFechaPers(Fecha fechaPers) {
61     FechaPers = fechaPers;
62 }
63
64 public char getGenePers() {
65     return GenePers;
66 }
```

```
eclipse-workspace - TemaClasesObjetos/src/TDA/Persona.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Cuadrado.java Rectangulo.java Triangulo.java Circulo.java TestFigurasGeometricas.java Persona2.java Persona.java
64 public char getGenePers() {
65     return GenePers;
66 }
67
68 public void setGenePers(char genePers) {
69     GenePers = genePers;
70 }
71 //Generacion del metodo toString
72
73 public String toString() {
74     return "Persona [Usted se llama: "+NomPers+". Su Numero es: "+TelPers+", con: "+EdadPers+"
75     +NacionPers+" Tu fecha es: "+FechaPers+" y usted es: "+GenePers+"]";
76 }
77
78
79 }
80
81
82
83
84
85
```

Detalles: Podemos observar que el código posee atributos privados los cuales son parametrizados en los Constructores, este código solo recibe como parámetros el nombre, edad, fecha de nacimiento, género y nacionalidad de una persona y lo almacena en la cadena o método toString para poder imprimir los datos. Ingresados de una persona



Fecha.

```
eclipse-workspace - TemaClasesObjetos/src/TDA/Fecha.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Persona2.java *Persona.java *Fecha.java X
1 package TDA;
2
3 public class Fecha {
4
5     private byte dia;
6     private byte mes;
7     private short año;
8
9     public Fecha () {}
10
11     public Fecha (byte dia,byte mes,short año) {
12
13         this.dia=dia;
14         this.mes=mes;
15         this.año=año;
16     }
17
18     public byte getDia() {
19         return dia;
20     }
21
22     public void setDia(byte dia) {
```

```
eclipse-workspace - TemaClasesObjetos/src/TDA/Fecha.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Persona2.java *Persona.java *Fecha.java X
23         this.dia = dia;
24     }
25
26     public byte getMes() {
27         return mes;
28     }
29
30     public void setMes(byte mes) {
31         this.mes = mes;
32     }
33
34     public short getAño() {
35         return año;
36     }
37
38     public void setAño(short año) {
39         this.año = año;
40     }
41
42     public String toString() {
43         return dia+" del "+mes+" del año: "+año;
44     }
45 }
```

Detalles: En este código se reciben como parámetros el número del día, mes y año para posteriormente con el método toString poder armar la cadena con el texto de la fecha y poderlo mostrar como mensaje en el código de Persona



## TestPersona.

```
eclipse-workspace - TemaClasesObjetos/src/TestClasesObjetos/TestPersona.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

1 package TestClasesObjetos;
2
3 import EntradasSalidas.Tools;
4
5
6
7 public class TestPersona {
8     public static void main(String[] args) {
9
10         capturaObjetos();
11     }
12     public static void capturaObjetos() {
13
14         String listado=" ";
15         int res=0;
16         byte contHom=0,contMuj=0,mayor=0;
17         Persona aux=new Persona();
18
19         do {
20             Persona obj=new Persona();
21
22
23             obj.setNomPers(Tools.LeerString("Nombre"));
24             obj.setEdadPers(Tools.LeerByte("Edad"));
25             obj.setNacionPers(Tools.LeerString("Nacionalidad"));
26             obj.setTelPers(Tools.LeerLong("Telefono"));
27             obj.setGenePers(Tools.LeerChar("Hombre o Mujer"));
28             obj.setFechaPers(new Fecha(Tools.LeerByte("Dia"),Tools.LeerByte("Mes"),Tools.Leershort("Año")));
29             //Tools.Salida("Datos Capturados:"+obj.toString());
30             listado+="Datos:\n"+obj.toString()+"\n";
31
32         } while (res!=1);
33     }
34 }

Activar Windows
Ve a Configuración para activar Windows;
```

```
eclipse-workspace - TemaClasesObjetos/src/TestClasesObjetos/TestPersona.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

31
32 if (obj.getGenePers()=='H' || obj.getGenePers()=='h')
33     contHom++;
34 else
35     if (obj.getGenePers()=='M' || obj.getGenePers()=='m')
36         contMuj++;
37
38 if (obj.getEdadPers()>mayor) {
39     mayor=obj.getEdadPers();
40     aux=obj;
41 }
42
43 res= (int) Tools.seguirSino();
44 }
45 while (res!=1);
46 byte total=(byte) (contHom+contMuj);
47
48
49 Tools.Salida("Lista de Personas:\n"+listado+"Hay "+contHom+" Hombres\n"+contMuj+" Mujeres"
50 +" \nFrecuencia de hombres es: " +TestPersona.imprimeFrecuencia(contHom)+" \nla de mujeres es de: "
51 +TestPersona.imprimeFrecuencia(contMuj)+" \nEl porcentaje de Hombres es de: "
52 +((100*contHom)/total)+"% \nEl porcentaje de Mujeres es de: "+((100*contMuj)/total)+"%"
53 +" \nLa persona con Mayor edad es: "+aux.toString());
54
55
56 //Tools.Salida(listado+" el total es "+total+" de hombres "+contHom+" y de mujeres "+contMuj);
57 }
58 public static String imprimeFrecuencia (byte n) { //Ejercicio 15

Activar Windows
Ve a Configuración para activar Windows;
```



```
eclipse-workspace - TemaClasesObjetos/src/TestClasesObjetos/TestPersona.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Persona2.java Persona.java Fecha.java TestPersona.java
52 +((100*contHom)/total)+"%\nEl porcentaje de Mujeres es de: "+((100*co
53 +"\nLa persona con Mayor edad es: "+aux.toString());
54
55
56 //Tools.Salida(listado+" el total es "+total+" de hombres "+contHom+"
57 }
58 public static String imprimeFrecuencia (byte n) {
59
60 String cad="";
61 for (int i=1;i<=n; i++ ) {
62
63     cad+="*";
64 }
65 return cad;
66 }
67
68 }
69
Activar Windows
Ve a Configuración para activar Windows
```

Detalles: En este código ocupamos el método main para poder ejecutar nuestro programa, Hemos creado un método que es llamado acá mismo para ejecutarlo en este método hemos creado el objeto con la palabra obj=new y lo hemos metido en un ciclo el cual estará constantemente preguntando al usuario si desearía volver a ingresar datos para posteriormente irlos anexando a una cadena el cual guardara los datos ingresados de los usuarios todas las veces que se desee, las entradas de datos se almacenan en los métodos sets

Además de que después de pedir los datos, estos mismo entran en una condición el cual compara la edad dada por el usuario cada vuelta y la registra en una variable el cual es el valor mayor de la edad, este mismo se muestra en la salida de datos como la persona con mayor edad el cual la indica.

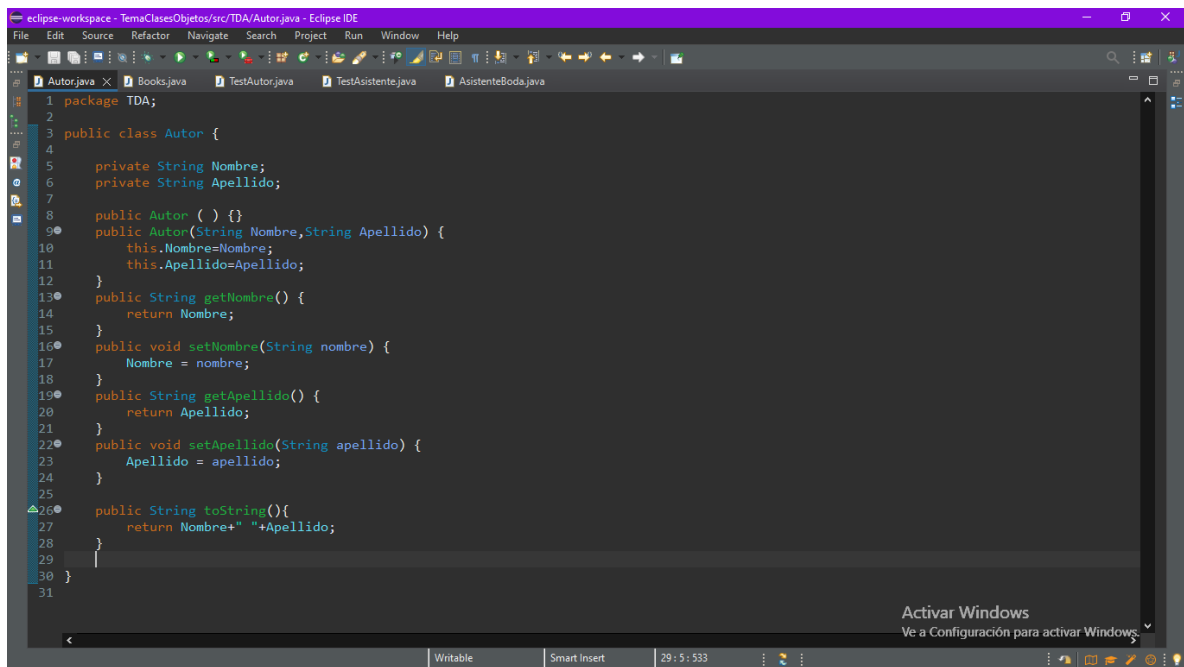
A su vez cuenta el porcentaje de mujeres y de hombres mediante la operación de 100 multiplicado por el número de mujeres/hombre y el resultado dado se divide por la sumatoria de mujeres y hombres mostrando, también se ha reutilizado un código el cual este muestra como mensaje una cierta cantidad de asteriscos (\*), se ha usado para imprimir cuantos hombres y mujeres se han ingresado en forma de asteriscos.



## Ejercicio 2.

### Orden del problema 2

#### Clase Autor.



```
1 package TDA;
2
3 public class Autor {
4
5     private String Nombre;
6     private String Apellido;
7
8     public Autor ( ) {}
9     public Autor(String Nombre,String Apellido) {
10         this.Nombre=Nombre;
11         this.Apellido=Apellido;
12     }
13     public String getNombre() {
14         return Nombre;
15     }
16     public void setNombre(String nombre) {
17         Nombre = nombre;
18     }
19     public String getApellido() {
20         return Apellido;
21     }
22     public void setApellido(String apellido) {
23         Apellido = apellido;
24     }
25
26     public String toString(){
27         return Nombre+" "+Apellido;
28     }
29 }
30
31
```

Detalles:Este programa recibe un tipo de dato llamado Nombre y otro llamado apellido los cuales al ser ingresados por el usuario en la unidad ejecutable estos mostraran la linea de texto dada en el metodo toString

#### Clase Books(Libros).





```
eclipse-workspace - TemaClasesObjetos/src/TDA/Books.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Autor.java Books.java TestAutor.java TestAsistente.java AsistenteBoda.java
1 package TDA;
2
3 public class Books {
4
5     private String Titulo;
6     private Autor Autor;
7     private int ISBN;
8     private short NumPag;
9
10    public Books () {}
11    public Books (String Titulo,Autor Autor,int ISBN,short NumPag) {
12
13        this.Titulo=Titulo;
14        this.Autor=Autor;
15        this.ISBN=ISBN;
16        this.NumPag=NumPag;
17    }
18    public String getTitulo() {
19        return Titulo;
20    }
21    public void setTitulo(String titulo) {
22        Titulo = titulo;
23    }
24    public Autor getAutor() {
25        return Autor;
26    }
27
28    }
29
30    }
31
32    }
33
34    }
35
36    }
37
38    }
39
40    }
41
42    }
43
44    }
45
46    }
47
48    }
49
50    }
51
52    }
53
54    }
55
56    }
57
58    }
59
60    }
61
62    }
63
64    }
65
66    }
67
68    }
69
70    }
71
72    }
73
74    }
75
76    }
77
78    }
79
80    }
81
82    }
83
84    }
85
86    }
87
88    }
89
90    }
91
92    }
93
94    }
95
96    }
97
98    }
99
100   }
```

```
eclipse-workspace - TemaClasesObjetos/src/TDA/Books.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Autor.java Books.java TestAutor.java TestAsistente.java AsistenteBoda.java
26 }
27 public void setAutor(Autor autor) {
28     Autor = autor;
29 }
30 public int getISBN() {
31     return ISBN;
32 }
33 public void setISBN(int isbn) {
34     ISBN = isbn;
35 }
36 public short getNumPag() {
37     return NumPag;
38 }
39 public void setNumPag(short numPag) {
40     NumPag = numPag;
41 }
42
43 public String toString() {
44     return "Titulo: "+Titulo+"\nDel autor "+Autor+"\n con ISBN: "+ISBN+"\n Con "+NumPag+" Paginas";
45 }
46
47
48 }
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100 }
```

Detalles:Esta clase recibe los datos ingresado por el usuario del libro que planea mostrar, recibiendo Autor,titulo del libro, ISBN y el numero de paginas, una vez dado esto los podemos mostrar en pantalla llamando al metodo toString.

TestAutor



```
eclipse-workspace - TemaClasesObjetos/src/TestClasesObjetos/TestAutor.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Books.java Cuadrado.java Rectangulo.java Triangulo.java Circulo.java Persona2.java TestAutor.java X
1 package TestClasesObjetos;
2
3 import EntradasSalidas.Tools;
4
5
6
7 public class TestAutor {
8
9     public static void main(String[] args) {
10         // TODO Auto-generated method stub
11
12         CapturaDatos();
13     }
14
15     public static void CapturaDatos () {
16
17         String Listado=" ",LMenor=" ";
18         int res=0;
19
20         Books LibroM=new Books ();
21
22         do {
23             Books obj=new Books();
24
25             obj.setAutor(new Autor(Tools.LeerString("Escriba el Nombre del Autor"),
26                                     Tools.LeerString("Escriba su apellido")));
27             obj.setTitulo(Tools.LeerString("Titulo:"));
28             obj.setISBN(Tools.LeerInt("ISBN"));
29             obj.setNumPag(Tools.Leershort("Paginas"));
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
262
```



Detalles: En este programa se ejecutará y probará nuestro código, ya que contiene el método main en este programa se encuentra alojado un método en el cual se han creado los objetos para la ejecución, por consiguiente se ha implementado un ciclo DO-WHILE el cual este nos pedirá en varias ocasiones los datos nuevamente hasta que el usuario decida cerrar el ciclo, en este mismo se ha invocado un método el cual está en el mismo método del objeto pero este es llamado para comparar si los objetos creados contiene un cierto número menor de hojas a comparación de los demás creados y por ende es mostrado en la ejecución como el libro con menos hojas.

### Ejercicio 3.

#### rentaAutos

```
1 package TDA;
2
3 public class rentaAutos {
4
5     private byte Dias;
6     private double KM;
7     private char Tipo;
8
9     public rentaAutos() {}
10    public rentaAutos(double KM, char Tipo, byte Dias) {
11
12        this.Dias=Dias;
13        this.KM=KM;
14        this.Tipo=Tipo;
15    }
16    public int getDias() {
17        return Dias;
18    }
19    public void setDias(byte dias) {
20        Dias = dias;
21    }
22    public double getKM() {
23        return KM;
24    }
25    public void setKM(double km) {
26        KM = km;
27    }
28    public char getTipo() {
29        return Tipo;
30    }
31    public void setTipo(char tipo) {
32        Tipo = tipo;
33    }
34 }
```



```
eclipse-workspace - TemaClasesObjetos/src/TDA/rentaAutos.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
AsistenteBoda.java Persona.java rentaAutos.java Fecha.java Triangulo.java TestPersona.java
34
35 public double detTarifa () {
36     double detTarifa=0;
37
38     switch (Tipo) {
39         case 'P':
40         case 'p':
41             detTarifa=200;
42             break;
43         case 'M':
44         case 'm':
45             detTarifa=350;
46             break;
47         case 'G':
48         case 'g':
49             detTarifa=450;
50             break;
51         default: return 0;
52     }
53
54     return detTarifa;
55 }
56
57 public double CalCosto () {
58     double Costo=0;
59
60
61     switch (Tipo) {
62         case 'P':
63         case 'p':
64             Costo=20*KM;
65             break;
66         case 'M':
```

```
eclipse-workspace - TemaClasesObjetos/src/TDA/rentaAutos.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
AsistenteBoda.java Persona.java rentaAutos.java Fecha.java Triangulo.java TestPersona.java
66     case 'M':
67     case 'm':
68         Costo=30*KM;
69         break;
70     case 'G':
71     case 'g':
72         Costo=40*KM;
73         break;
74     default: Costo=0;
75 }
76 return Costo;
77 }
78
79 public double CalMonto () {
80     double Monto=(detTarifa()*Dias)+CalCosto();
81     double Extra=0;
82
83     if (KM>10) {
84         Extra=Monto*2.5;
85         Monto+=Extra;
86     }
87     return Monto;
88 }
89 }
90
91 }
92
93
94
95
96
97
98
```

Detalles: En este programa recibe los datos ingresados por el usuario del objeto creado que son los Días que son los días en los se rentó el automóvil, KM que son los km recorridos, y el Tipo que es tipo del automóvil, ya una vez almacenados son ocupados para desarrollar los siguientes métodos



detTarifa: este método recibe el atributo llamado tipo y lo compara en base a los valores ya definidos en el método, una vez comparados y que sean iguales a uno de ellos regresa como resultado la tarifa del tipo de automóvil que concuerda.

detCosto: este método al igual que el anterior recibe como parámetro el atributo llamado Tipo y de igual manera lo compara con los valores ya definidos y en el caso de ser igual a uno regresa un precio, sin embargo, este resultado lo multiplica por el atributo llamado KM dando como resultado el valor del costo final.

CalMonto: este método crea como variable monto que es el resultado de la operación la multiplicación del método de detTarifa por Dias y al resultado se le multiplica por detCosto, una vez definido el Monto se evalúa si el dato ingresado en el atributo KM es mayor a 10, de cumplirse esta condición el resultado de Monto se multiplicara por 2.5 que es la cuota a pagar por la penalización de exceder el límite de 10, en caso contrario de que no se haya excedido el Monto queda como esta antes de la condición.

## TestRentaAutos

```
eclipse-workspace - TemaClasesObjetos/src/TestClasesObjetos/TestRentaAutos.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

1 package TestClasesObjetos;
2
3 import EntradasSalidas.Tools;
4
5
6 public class TestRentaAutos {
7
8     public static void main(String[] args) {
9         // TODO Auto-generated method stub
10
11         CapturaDatos();
12     }
13
14     public static void CapturaDatos () {
15
16         String List="Renta del Auto\n";
17         byte res=0;
18
19         rentaAutos auto=new rentaAutos();
20         do {
21             auto.setTipo(Tools.leerChar("Inserte el tipo de auto el cual puede ser:\n [P]equeno\n [M]ediano\n [G]rande"));
22             auto.setDias(Tools.leerByte("Ingrese los dias los cuales rentara el auto"));
23             auto.setKM(Tools.leerDouble("Ingrese la distancia que recorrio(KM):"));
24             List+="\nSu tipo de auto es: "+auto.getTipo()+"\n"+ "Los dias alquilados fueron: "+auto.getDias()+" dias\n"+
25             "Su distancia recorrida fue de: "+auto.getKM()+" KM\n"+ "Su tarifa fue de: "+auto.detTarifa()+"$\n"+
26             "El costo a pagar por la renta es de: "+auto.CalCosto()+"$\n"+
27             "Usted pagara en total: "+auto.CalMonto()+"$\n";
28             res= (byte) Tools.seguirSino();
29         }
30         while (res!=1);
31         Tools.Salida("Lista de Datos\n"+List);
32     }
33 }
```

Detalles: En este código es donde se ejecutará el código ya que posee el método main, se añadió las entradas de datos a los métodos sets y posteriormente se



implementó un ciclo el cual pedirá constantemente al usuario que vuelva a ingresarlos las veces que desee, concatenando los objetos creados en una cadena el cual almacena los datos para posteriormente imprimirlos.

Ejercicio 4.

Orden del Ejercicio 4.

Cuadrado.

```
1 package TDA;
2
3 public class Cuadrado {
4
5     private float Lado;
6
7     public Cuadrado() {}
8     public Cuadrado(float Lado) {
9
10        this.Lado=Lado;
11    }
12
13    public float getLado() {
14        return Lado;
15    }
16
17    public void setLado(float lado) {
18        Lado = lado;
19    }
20
21
22    public float Area () {
23
24        return (float) Math.pow(Lado,2);
25    }
26
27
28    public float Perimetro () {
29
30        return Lado*4;
31    }
32
33 }
```

Detalles: Este programa recibe un atributo que es Lado, posterior a contiene 2 métodos

Area: este método calcula el área de la figura, en este caso un Cuadrado, eleva el Lado al cuadrado dando como resultado el área.

Perímetro: este método calcula el perímetro del cuadrado utilizando la formula Lado\*4.



## Triangulo.

```
eclipse-workspace - TemaClasesObjetos/src/TDA/Triangulo.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
[Cuadrado.java] [Rectangulo.java] [Triangulo.java X] [Circulo.java] [TestFigurasGeometricas.java]
1 package TDA;
2
3 public class Triangulo {
4     //Angel de Jesus y Kevin Alan
5     private float Base;
6     private float Altura;
7
8     public Triangulo () {}
9     public Triangulo (float Base,float Altura) {
10
11         this.Base=Base;
12         this.Altura=Altura;
13     }
14
15     public float getBase() {
16         return Base;
17     }
18
19     public void setBase(float base) {
20         Base = base;
21     }
22
23     public float getAltura() {
24         return Altura;
25     }
26
27     public void setAltura(float altura) {
28         Altura = altura;
29     }
30
31     public float Hipotenusa3() {
32         double suma;
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2
```



Hipotenusa3: este método nos da el valor de la hipotenusa de triángulo formado, el cual es el lado más largo del triángulo, este se logra obtener por la fórmula de La suma de la base elevada al cuadrado más el lado elevado al cuadrado, al resultado de esta suma se le saca su raíz cuadrada y obtenemos la hipotenusa

Tipo Triangulo: este método compara los valores de los lados que posea el triángulo creado y los trata de igualar con ciertas condiciones como, por ejemplo, si todos los lados son iguales es un triángulo equilátero, si los lados son diferentes es un triángulo escaleno y si dos son iguales y uno diferente se considera un triángulo equilátero

Perímetro: Calcula el perímetro del triángulo el cual es la suma de todos sus lados

Area: Calcula el área del triángulo utilizando la formula  $\text{Base} \times \text{Altura}$  y el resultado se divide entre dos.

Rectángulo.

```
eclipse-workspace - TemaClasesObjetos/src/TDA/Rectangulo.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
*Cuadrado.java *Rectangulo.java *Triangulo.java Circulo.java TestFigurasGeometricas.java
1 package TDA;
2
3 public class Rectangulo {
4
5     private float Altura;
6     private float Base;
7
8     public Rectangulo() {}
9
10    public Rectangulo(float Altura, float Base) {
11
12        this.Altura=Altura;
13        this.Base=Base;
14    }
15
16    public float getAltura() {
17        return Altura;
18    }
19
20    public void setAltura(float altura) {
21        Altura = altura;
22    }
23 }
```





```
eclipse-workspace - TemaClasesObjetos/src/TDA/Rectangulo.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
*Cuadrado.java *Rectangulo.java *Triangulo.java Circulo.java TestFigurasGeometricas.java
22 }
23
24 public float getBase() {
25     return Base;
26 }
27
28 public void setBase(float base) {
29     Base = base;
30 }
31
32 public float Perimetro () {
33
34     return 2*(Altura+Base);
35 }
36
37 public float Area() {
38
39     return Altura*Base;
40 }
41
42 }
43 }
```

Detalles: Este código se podría definir como una mezcla del triángulo y cuadrado ya que de igual manera recibe dos atributos llamados Base y Altura, estos dos atributos conforman lo que es la figura geométrica el cual es el triángulo, y de igual manera posee dos métodos

Perímetro: este método nos regresa el perímetro de nuestra figura el cual es Altura sumada a la base y el resultado multiplicado por dos

Area: este método nos regresa el área el cual es obtenido por la operación de altura multiplicada por base.

Circulo



```
1 package TDA;
2
3 public class Circulo {
4
5     private float radio;
6
7     public Circulo() {}
8     public Circulo(float radio) {
9
10        this.radio=radio;
11    }
12
13    public float getRadio() {
14        return radio;
15    }
16
17    public void setRadio(float radio) {
18        this.radio = radio;
19    }
20
21    public float Perimetro() {
22
23        return (float) (2*(Math.PI*radio));
24    }
25
26    public float Area() {
27        float raiz;
28
29        raiz=(float) Math.pow(radio,2);
30        return (float) (Math.PI*raiz);
31    }
32 }
33
```

Detalles: El código pose el atributo llamado Radio, el cual nos ayudara a definir los siguientes métodos

Perímetro: el cual regresa el perímetro del círculo mediante la fórmula que es el radio multiplicado por PI () y el resultado multiplicado por 2 no da el valor de su perímetro

Area: Este regresa el área del círculo obtenida de la formula el cual dice que el radio se eleva al cuadrado y el resultado se multiplica por PI () siendo el resultado el area

TestFigurasGeometricas.



```
eclipse-workspace - TemaClasesObjetos/src/TestClasesObjetos/TestFigurasGeometricas.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
*Cuadrado.java *Rectangulo.java *Triangulo.java *Circulo.java *TestFigurasGeometricas.java X
1 package TestClasesObjetos;
2
3 import javax.swing.JOptionPane;
4
5 import EntradasSalidas.Tools;
6 import TDA.Circulo;
7 import TDA.Cuadrado;
8 import TDA.Rectangulo;
9 import TDA.Triangulo;
10
11 public class TestFigurasGeometricas {
12
13     public static void main(String[] args) {
14         // TODO Auto-generated method stub
15
16         TestFigurasGeometricas.Menu("Triangulo,Cuadrado,Circulo,Rectangulo,Salir");
17     }
18 String menu="Triangulo,Cuadrado,Circulo,Rectangulo,Salir";
19     public static void Menu(String menu) {
20         String sel;
21         Triangulo obj=new Triangulo();
22         Cuadrado Lad=new Cuadrado();
23         Circulo Rad=new Circulo();
24         Rectangulo Cara=new Rectangulo();
25
26         do {
27             sel= despegable(menu);
28             switch(sel) {
29                 case "Triangulo":obj.setAltura(Tools.LeerFloat("Ingrese la altura del triangulo"));
30                 obj.setBase(Tools.LeerFloat("Ingrese la base del Triangulo"));
31                 Tools.Salida("El triangulo es de tipo "+obj.TipoTriangulo()+"\n"+"Su hipotenusa
32                 obj.Hipotenusa3()+" M^3+"\n"+"Su Perimetro es de: "+obj.Perimetro()+" M^3"+
33                 "\n"+"Su Area de "+obj.Area()+" M^3");
34                 break;
35                 case "Cuadrado": Lad.setLado(Tools.LeerFloat("Ingrese el valor del Lado"));
36                 Tools.Salida("El cuadrado tiene como perimetro: "+Lad.Perimetro()+"\n"+"Su area
37                 break;
38                 case "Rectangulo": Cara.setAltura(Tools.LeerFloat("Ingrese la Altura"));
39                 Cara.setBase(Tools.LeerFloat("Ingrese la Base"));
40                 Tools.Salida("El Perimetro del Rectangulo es: "+Cara.Perimetro()+"\n"+"La Area
41                 break;
42                 case "Circulo":
43                     Rad.setRadio(Tools.LeerFloat("Ingrese el Radio del circulo"));
44                     Tools.Salida("El circulo tiene como perimetro: "+Rad.Perimetro()+"\n"+"Su area
45                     break;
46                 case "Salir":
47                     return;
48             }
49         } while (sel != "Salir");
50     }
51
52     private static String despegable(String menu) {
53         String sel;
54         do {
55             sel= JOptionPane.showInputDialog("Seleccione una opcion: "+menu);
56         } while (sel == null || sel.length() == 0);
57         return sel;
58     }
59 }
```

```
eclipse-workspace - TemaClasesObjetos/src/TestClasesObjetos/TestFigurasGeometricas.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
*Cuadrado.java *Rectangulo.java *Triangulo.java *Circulo.java *TestFigurasGeometricas.java X
23 Circulo Rad=new Circulo();
24 Rectangulo Cara=new Rectangulo();
25
26
27 do {
28     sel= despegable(menu);
29     switch(sel) {
30         case "Triangulo":obj.setAltura(Tools.LeerFloat("Ingrese la altura del triangulo"));
31         obj.setBase(Tools.LeerFloat("Ingrese la base del Triangulo"));
32         Tools.Salida("El triangulo es de tipo "+obj.TipoTriangulo()+"\n"+"Su hipotenusa
33         obj.Hipotenusa3()+" M^3+"\n"+"Su Perimetro es de: "+obj.Perimetro()+" M^3"+
34         "\n"+"Su Area de "+obj.Area()+" M^3");
35         break;
36         case "Cuadrado": Lad.setLado(Tools.LeerFloat("Ingrese el valor del Lado"));
37         Tools.Salida("El cuadrado tiene como perimetro: "+Lad.Perimetro()+"\n"+"Su area
38         break;
39         case "Rectangulo": Cara.setAltura(Tools.LeerFloat("Ingrese la Altura"));
40         Cara.setBase(Tools.LeerFloat("Ingrese la Base"));
41         Tools.Salida("El Perimetro del Rectangulo es: "+Cara.Perimetro()+"\n"+"La Area
42         break;
43         case "Circulo":
44             Rad.setRadio(Tools.LeerFloat("Ingrese el Radio del circulo"));
45             Tools.Salida("El circulo tiene como perimetro: "+Rad.Perimetro()+"\n"+"Su area
46             break;
47         case "Salir":
48             return;
49     }
50 } while (sel != "Salir");
51
52 private static String despegable(String menu) {
53     String sel;
54     do {
55         sel= JOptionPane.showInputDialog("Seleccione una opcion: "+menu);
56     } while (sel == null || sel.length() == 0);
57     return sel;
58 }
```



```
eclipse-workspace - TemaClasesObjetos/src/TestClasesObjetos/TestFigurasGeometricas.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
...
40      Cara.setBase(Tools.LeerFloat("Ingrese la Base"));
41      Tools.Salida("El Perimetro del Rectangulo es: "+Cara.Perimetro()+"\n"+"La Area ");
42      break;
43      case "Circulo":
44          Rad.setRadio(Tools.LeerFloat("Ingrese el Radio del circulo"));
45          Tools.Salida("El perimetro del circulo es de: "+Rad.Perimetro()+"\n"+"El Ar");
46          break;
47      case "Salir": break;
48      }
49
50      }while(!sel.equalsIgnoreCase("Salir"));
51  }
52
53  public static String despegable(String menu) {
54      String valores[]=menu.split(",");
55      String res= (String)JOptionPane.showInputDialog(null,"MENU","Selecciona la opcion:");
56
57      return (res);
58  }
59  }
60 }
61
```

Detalles: Este código se encuentra en la unidad ejecutable para su correcta evaluación, el método que ejecuta contiene la creación de 4 objetos correspondientes a las figuras geométricas, posteriormente se ha creado un menú que contiene estas figuras y el cual es llamado para presentarnos las opciones a elegir para acceder a las clases de estas figuras y hacer las operaciones debidas, hasta que el usuario desee salir.

Ejercicio 5.

Persona2.



```
eclipse-workspace - TemaClasesObjetos/src/TDA/Persona2.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
*Persona2.java x Persona.java Fecha.java TestPersona.java
1 package TDA;
2
3 public class Persona2 {
4
5     private String Nombre;
6     private byte Edad;
7     private char Sexo;
8     private float Peso;
9     private float Altura;
10
11 public Persona2() {}
12
13 public Persona2(String Nombre,byte Edad,char Sexo,float Peso,float Altura) {
14
15     this.Altura=Altura;
16     this.Peso=Peso;
17     this.Nombre=Nombre;
18     this.Edad=Edad;
19     this.Sexo=Sexo;
20 }
21
22 public Persona2(String Nombre,byte Edad,char Sexo) {
23
24     this.Edad=Edad;
25     this.Nombre=Nombre;
26     this.Sexo=Sexo;
27
28 }
29
30 }
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
26
```



```
eclipse-workspace - TemaClasesObjetos/src/TDA/Persona2.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
*Persona2.java x Persona.java Fecha.java TestPersona.java
53     return Altura;
54 }
55 public void setAltura(float altura) {
56     Altura = altura;
57 }
58
59
60 public byte calcularMC() {
61     byte n=0;
62     float pi= (float) (Peso/(Math.pow(Altura, 2)));
63
64     switch ((pi<20)?1:(pi>=20 && pi<=25)?2:(pi>25)?3:4) {
65
66     case 1:n=-1;
67     break;
68     case 2:n=0;
69     break;
70     case 3:n=1;
71     break;
72
73     }
74
75     return n;
76 }
77
78 public boolean esMayorDeEdad() {
```

```
eclipse-workspace - TemaClasesObjetos/src/TDA/Persona2.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
*Persona2.java x Persona.java Fecha.java TestPersona.java
73     }
74
75     return n;
76 }
77
78 public boolean esMayorDeEdad() {
79
80     if (Edad>18) {
81         return true;
82     }
83     else return false ;
84
85 }
86
87 public String toString() {
88
89     return "Su nombre es: "+Nombre+"\n"+"Tiene: "+Edad+" años"+" \n"+"Usted es "+Sexo
90         +"\n"+"Usted pesa "+Peso+"KG"+" \n"+"Usted Mide "+Altura+"m";
91 }
92
93 public String ToString() {
94
95     return "Su nombre es: "+Nombre+"\n"+"Tiene: "+Edad+" años"+" \n"+"Usted es "+Sexo+"\n";
96 }
97 }
98
```

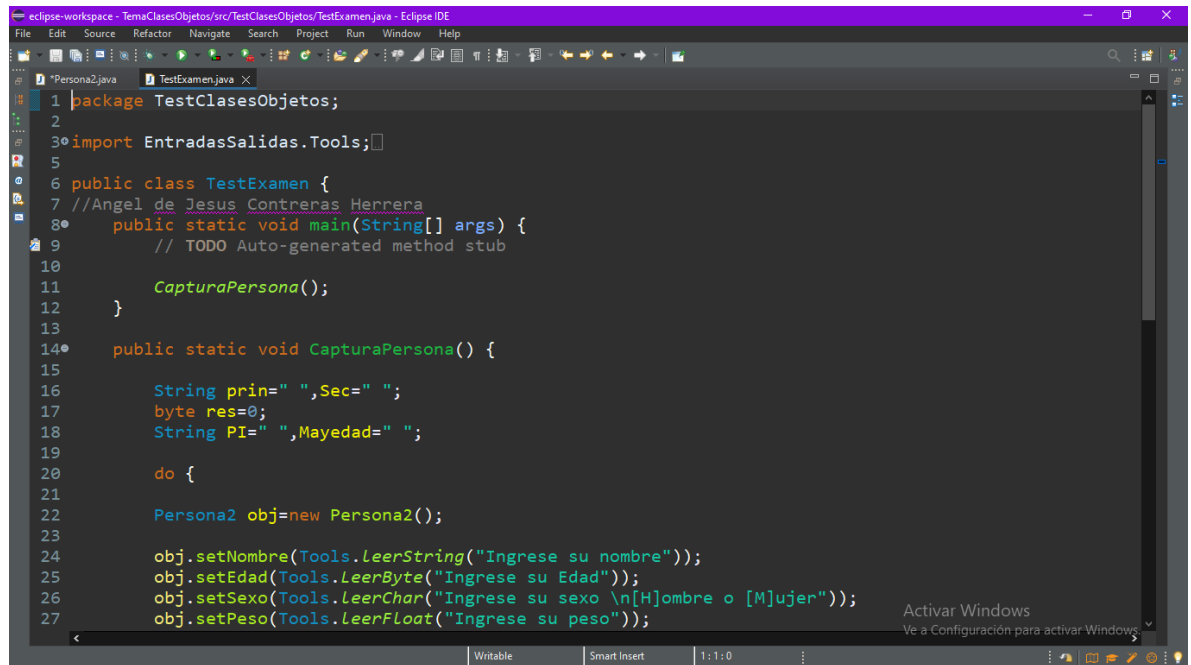
Detalles: El Código puede recibir 5 atributos los cuales son Nombre, edad, sexo, peso y altura de una persona, lo cuales son todos almacenados en un método llamado toString el cual puede imprimir todos datos en la pantalla y otro llamado ToString el cual solo imprime 3 datos en la pantalla el cual es nombre, edad y sexo de la persona, a su vez este posee 2 métodos los cuales son:



CalcularMC este método eleva la altura de la persona al cuadrado y el resultado lo divide por el peso, una vez resuelto esta operación el resultante se usa para comparar con valores ya definidos los cuales conforme a la similitud de uno de ellos este devuelve un valor el cual puede ser -1, 0 y 1 que se usara para una futura comparación.

esMayorDeEdad: este método compara si la edad ingresada es mayor a 18, de cumplirse esta condición este método devuelve un valor booleano verdadero/true o falso/false.

TestExamen.



```
1 package TestClasesObjetos;
2
3 import EntradasSalidas.Tools;
4
5
6 public class TestExamen {
7     //Angel de Jesus Contreras Herrera
8     public static void main(String[] args) {
9         // TODO Auto-generated method stub
10
11         CapturaPersona();
12     }
13
14     public static void CapturaPersona() {
15
16         String prin=" ",Sec=" ";
17         byte res=0;
18         String PI=" ",Mayedad=" ";
19
20         do {
21
22             Persona2 obj=new Persona2();
23
24             obj.setNombre(Tools.LeerString("Ingrese su nombre"));
25             obj.setEdad(Tools.LeerByte("Ingrese su Edad"));
26             obj.setSexo(Tools.LeerChar("Ingrese su sexo \n[H]ombre o [M]ujer"));
27             obj.setPeso(Tools.LeerFloat("Ingrese su peso"));
```



```
eclipse-workspace - TemaClasesObjetos/src/TestClasesObjetos/TestExamen.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

28 obj.setAltura(Tools.LeerFloat("Ingrese su altura"));
29
30 if (obj.calcularMC()==-1) {
31     PI="Está usted por debajo de su peso ideal";
32 }
33
34 else if (obj.calcularMC()==0) {
35     PI="Está usted en su peso ideal";
36 }
37 else
38     PI="Usted tiene Sobrepeso";
39
40 if (obj.esMayorDeEdad()==true)
41     Mayedad="Usted es mayor de Edad";
42 else
43     Mayedad="Usted no es mayor de Edad";
44
45 prin="\nDatos:\n "+obj.toString()+"\n"+PI+"\n"+Mayedad+"\n";
46
47 Persona2 oct=new Persona2();
48
49 oct.setNombre(obj.getNombre());
50 oct.setEdad(obj.getEdad());
51 oct.setSexo(obj.getSexo());
52
53
Activar Windows
Ve a Configuración para activar Windows
```

```
eclipse-workspace - TemaClasesObjetos/src/TestClasesObjetos/TestExamen.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

38 PI="Usted tiene Sobrepeso";
39
40 if (obj.esMayorDeEdad()==true)
41     Mayedad="Usted es mayor de Edad";
42 else
43     Mayedad="Usted no es mayor de Edad";
44
45 prin="\nDatos:\n "+obj.toString()+"\n"+PI+"\n"+Mayedad+"\n";
46
47 Persona2 oct=new Persona2();
48
49 oct.setNombre(obj.getNombre());
50 oct.setEdad(obj.getEdad());
51 oct.setSexo(obj.getSexo());
52
53
54 Sec+=prin+"\nDatos:\n"+oct.ToString();
55
56 res= (byte) Tools.seguirSino();
57 }
58 while (res!=1);
59
60 Tools.Salida("Informacion:"+Sec);
61 }
62 }
63
Activar Windows
Ve a Configuración para activar Windows
```

Detalles: El código contiene la unidad ejecutable, a su vez el método con la creación del objeto, dentro de este método hemos de crear un objeto en un ciclo que estará constantemente preguntando al usuario si desea ingresar nuevos datos o terminar el ciclo, posterior a esto se han añadido una serie de condiciones que determinaran si una persona está en su peso ideal o no, mediante el método





CalcularMC retornara un valor el cual puede ser -1, 0 y 1, y los valores definidos son los siguientes: -1 = Usted está por debajo de su peso ideal, 0 = usted está en su peso ideal y 1 = usted tiene sobrepeso.

Una vez cumplida estas condiciones se ha implementado otra condición en referencia a la edad el cual utiliza el método esMayorDeEdad que retornará true o false según el criterio se cumpla, en el caso de cumplirse la condición true regresará un mensaje diciendo que la persona es mayor de edad, en caso contrario regresará que es menor de edad, todo concatenado en una cadena que almacenara los datos de los ciclos dados por el usuario, además, se ha creado un segundo objeto con la característica que solo pedirá nombre, edad y sexo de la persona y los datos lo regresará en una cadena y este objeto ocupara el método ToString el cual contiene los datos para mostrar con esta característica.

## Conclusión:

### Ejercicio 1.

Mensaje

?

Lista de Personas:

Datos:

Persona [Usted se llama: Angel de Jesus. Su Numero es: 2713151250, con: 20 años Nacionalidad: Mexicana Tu fecha es: 2 del 3 del año: 2003 y usted es: H]

Datos:

Persona [Usted se llama: Patricia. Su Numero es: 2711901070, con: 50 años Nacionalidad: Mexicana Tu fecha es: 28 del 5 del año: 1970 y usted es: M]

Datos:

Persona [Usted se llama: Rosa. Su Numero es: 2711063663, con: 70 años Nacionalidad: Mexicana Tu fecha es: 26 del 7 del año: 1950 y usted es: M]

Datos:

Persona [Usted se llama: Miguel. Su Numero es: 2712093287, con: 48 años Nacionalidad: Mexicana Tu fecha es: 10 del 3 del año: 1968 y usted es: H]

Hay 2 Hombres

2 Mujeres

Frecuencia de hombres es: \*\*

la de mujeres es de: \*\*

El porcentaje de Hombres es de: 50%

El porcentaje de Mujeres es de: 50%

La persona con Mayor edad es: Persona [Usted se llama: Rosa. Su Numero es: 2711063663, con: 70 años Nacionalidad: Mexicana Tu fecha es: 26 del 7 del año: 1950 y usted es: M]

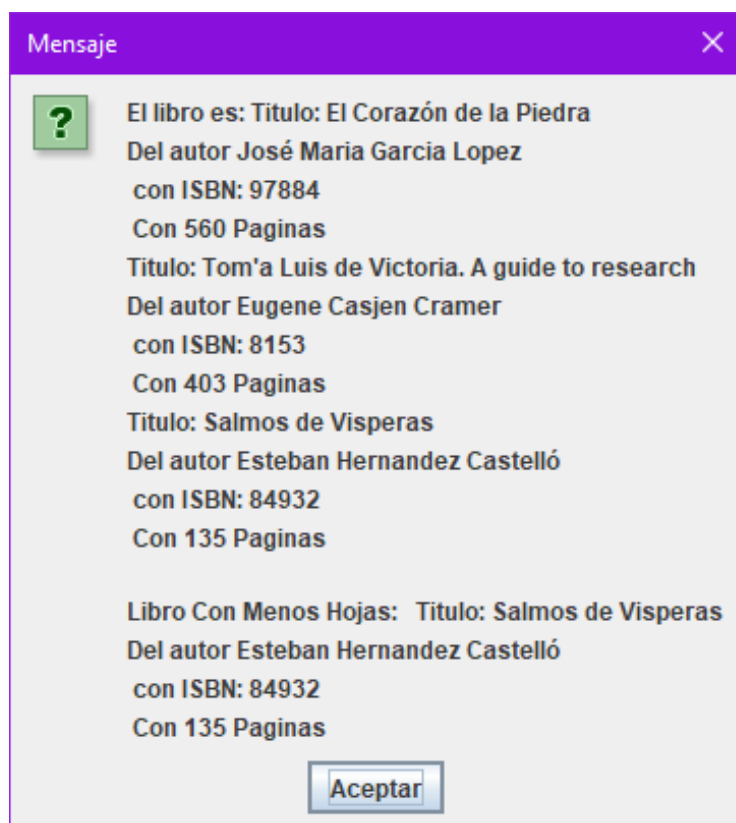
Aceptar

La impresión del código nos da un resultado el cual se ha impreso los datos de los usuarios ingresados, después se han contabilizado el total de hombre y mujeres según los datos obtenidos para así lograr sacar la frecuencia del hombre y de la mujer de igual manera su porcentaje que abarcan cada uno, también podemos



observar que ha logrado mostrar la persona con mayor edad de todos los ingresados, en conclusión, el código ha realizado lo previsto al resultado esperado.

## Ejercicio 2.



Como podemos observar en la ventana emergente resultado del código, no ha concatenado los datos de manera correcta los cuales fueron ingresados por el usuario, los títulos, autores, ISBN y el número de páginas ha sido mostrados correctamente, estos mismos han sido comparados entre sí para poder obtener el libro que contenga el menor números de hojas y mostrarlo en pantalla, por lo que en conclusión podemos decir que el código cumple con su propósito y si ha llegado a los resultados esperados.



### Ejercicio 3.

Mensaje

?

Lista de Datos

Renta del Auto

Su tipo de auto es: P

Los dias alquilados fueron: 10 dias

Su distancia recorrdia fue de: 50.0 KM

Su tarifa fue de: 200.0\$

El costo a pagar por la renta es de: 1000.0\$

Usted pagara en total: 10500.0\$

Su tipo de auto es: M

Los dias alquilados fueron: 30 dias

Su distancia recorrdia fue de: 20.0 KM

Su tarifa fue de: 350.0\$

El costo a pagar por la renta es de: 600.0\$

Usted pagara en total: 38850.0\$

Su tipo de auto es: G

Los dias alquilados fueron: 25 dias

Su distancia recorrdia fue de: 35.0 KM

Su tarifa fue de: 450.0\$

El costo a pagar por la renta es de: 1400.0\$

Usted pagara en total: 44275.0\$

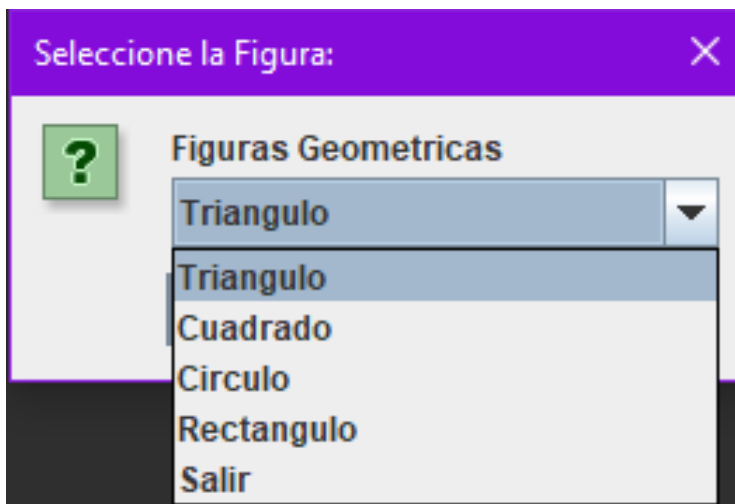
Aceptar



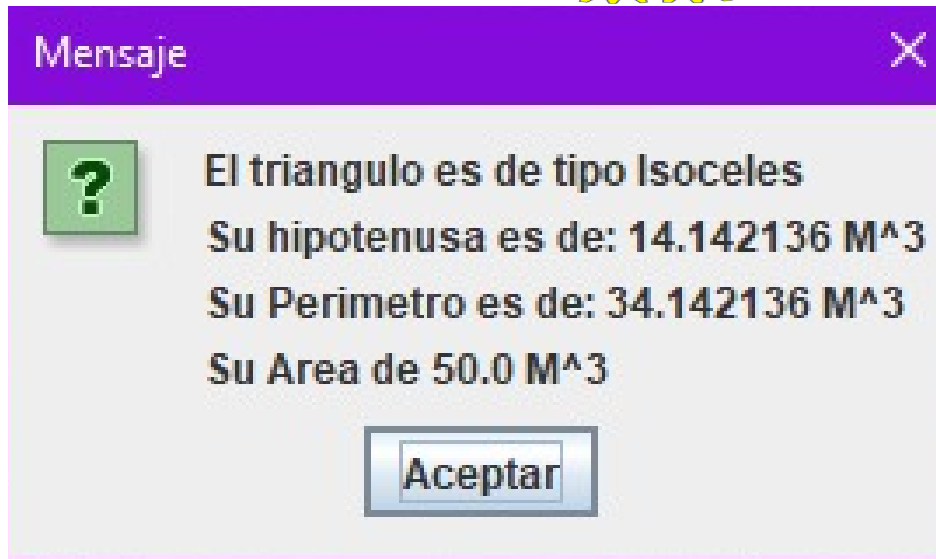
La ventana nos muestra los datos de los alquileres registrados por una empresa de autos, los cuales según el tipo de auto disponible se han determinado la tarifa, su costo por la distancia recorrida y el monto a pagar por el alquiler del vehículo, por lo que se ha concluido que el código cumple su propósito de proporcionar los montos a pagar de los clientes de una empresa.

#### Ejercicio 4

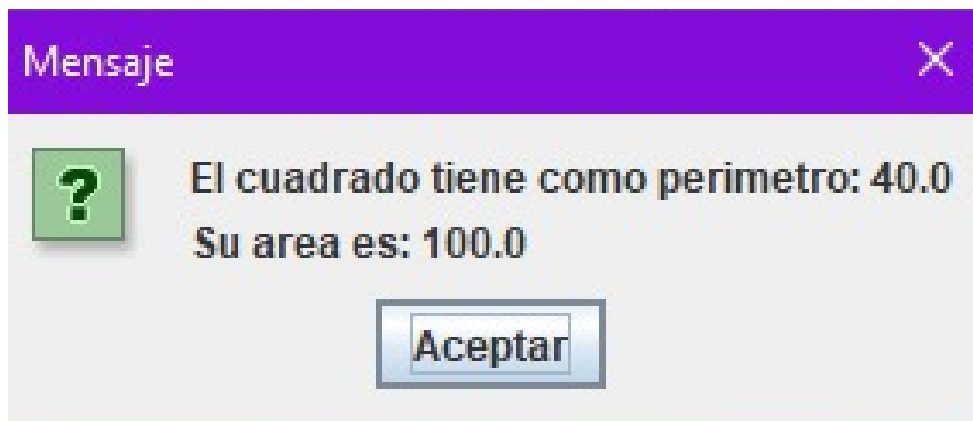
Prueba del Código.



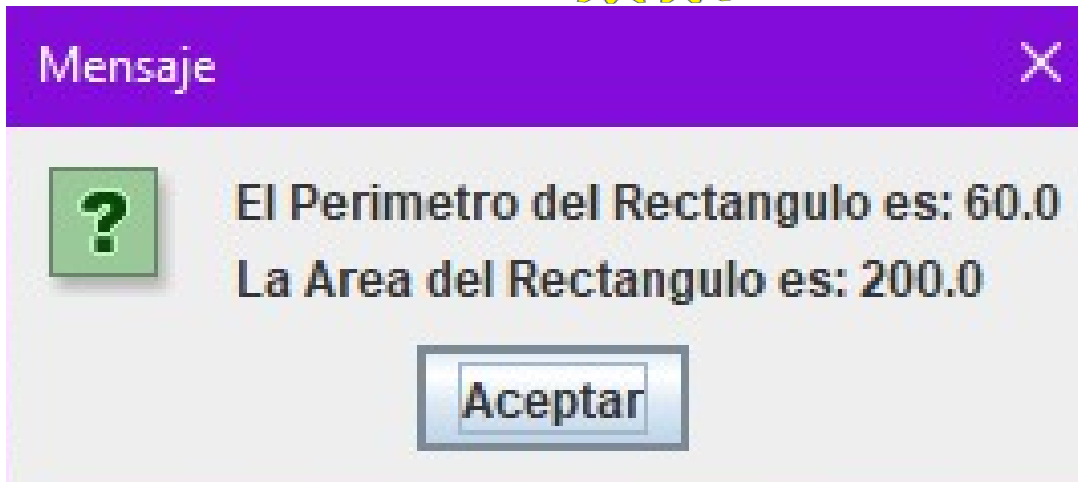
Triangulo.



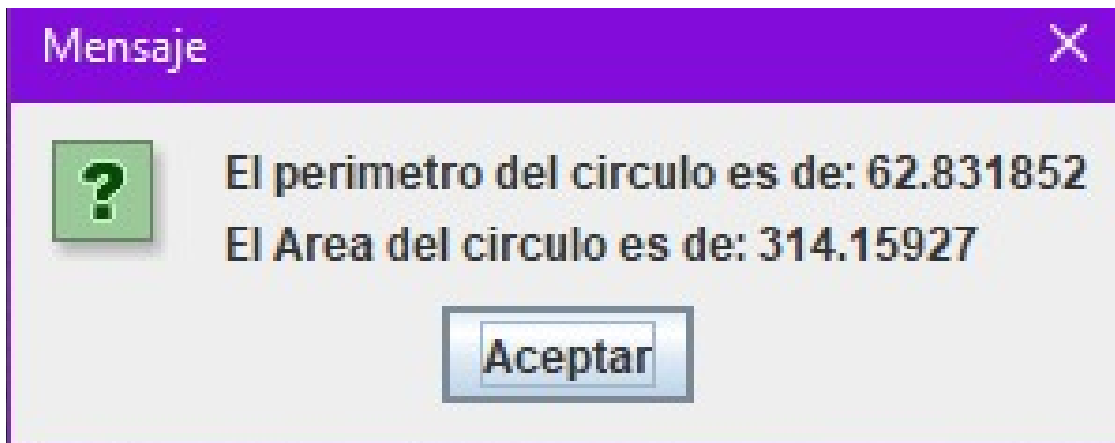
Cuadrado.



Rectángulo.



Circulo.



Las siguientes ventanas resultados de los códigos ejecutados no han dado la información de sus comportamientos, como podemos observar se trata de un menú el cual el usuario ha podido seleccionar su figura e ingresar las medidas para obtener sus respectivos valores según la figura seleccionada,

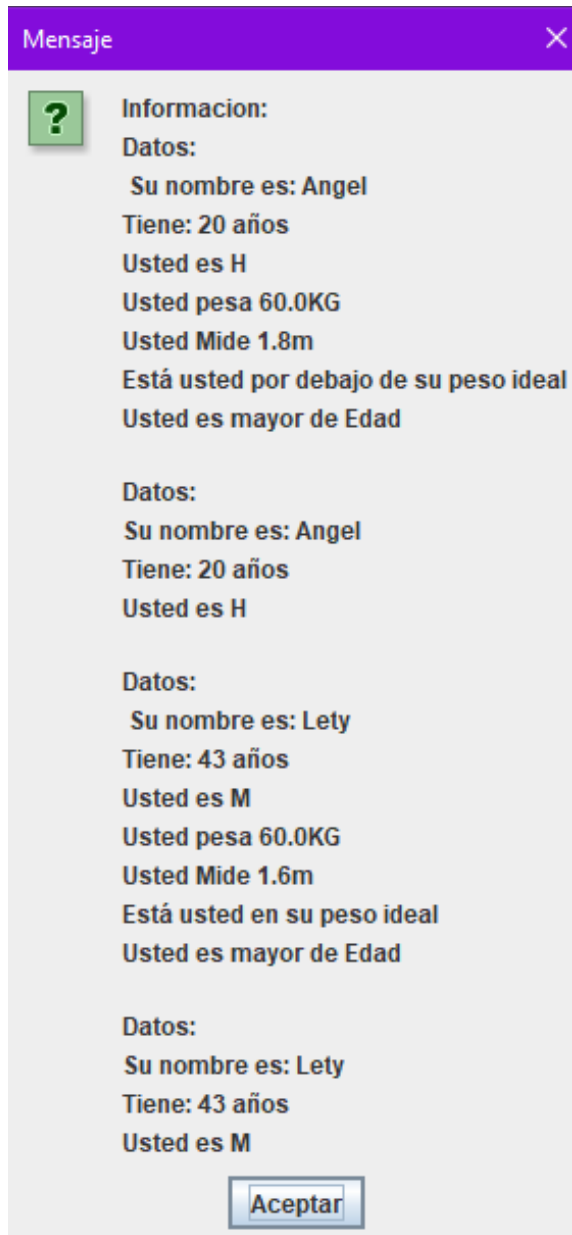
En el caso del triángulo, ha podido obtener de manera correcta la hipotenusa, debido a que se ha obtenido la hipotenusa se ha podido clasificar que tipo de triangulo es, isósceles, escaleno o equilátero y por ende el área y el perímetro son correctos, al igual que las demás figuras, sus debidos cálculos de sus respectivas áreas y perímetros son correctos.



TECNOLÓGICO  
NACIONAL DE MÉXICO



Ejercicio 5.



Y

Como hemos visto, este es el resultado del código que se ha ejecutado y como podemos ver ha concatenado los datos de las personas de manera correcta y ha logrado definir si la persona se encuentra en su peso ideal con base a su peso y





altura, de misma ha logrado mostrar si la persona es mayo o no con base a los datos de las edades de las personas

También se había solicitado volver a imprimir los datos, pero a excesion del peso y altura de la persona, por lo cual el código ha cumplido su funcionamiento correctamente

## Referencias

GARRO, A. (15 de Abril de 2014). *Java*. Obtenido de <https://www.arkaitzgarro.com/java/capitulo-9.html>

GERVAIS, L. (Mayo de 2019). *Aprender la Programación Orientada a Objetos con el lenguaje Java*. Obtenido de <https://www.ediciones-eni.com/open/mediabook.aspx?idR=8801ef69f950a528c974d8a3ca48327a>

González, J. D. (2022). Obtenido de ProgramarYa: <https://www.programarya.com/Cursos/Java/Objetos-y-Clases>

Robledo, L. A. (22 de Marzo de 2014). *Blogger*. Obtenido de Programacion Orientada a Objetos UNIDAD III: <https://luisfcb243.blogspot.com/2014/03/programacion-orientada-objetos-unidad.html>

Rodriguez, G. J. (18 de Febrero de 2012). *.Northware*. Obtenido de <https://www.northware.mx/blog/programacion-orientada-a-objetos/#:~:text=Los%20miembros%20de%20una%20clase,que%20est%C3%A1%20hecho%20un%20objeto.>

Unknown. (Mayo de 2013). *Blogger*. Obtenido de This is Java!!!!: <https://chilejava.blogspot.com/2013/05/especificadores-de-acceso.html>

Victoria, T. L. (s.f.). *Libros*. Obtenido de <https://www.uma.es/victoria/bibliografia.html>

Villalobos, J. A., & Casallas, R. (s.f.). Obtenido de [https://universidad-de-los-andes.gitbooks.io/fundamentos-de-programacion/content/Nivel4/5\\_ManejoDeLasExcepciones.html](https://universidad-de-los-andes.gitbooks.io/fundamentos-de-programacion/content/Nivel4/5_ManejoDeLasExcepciones.html)



TECNOLÓGICO  
NACIONAL DE MÉXICO

