



Lucas Varela Negro Dam1

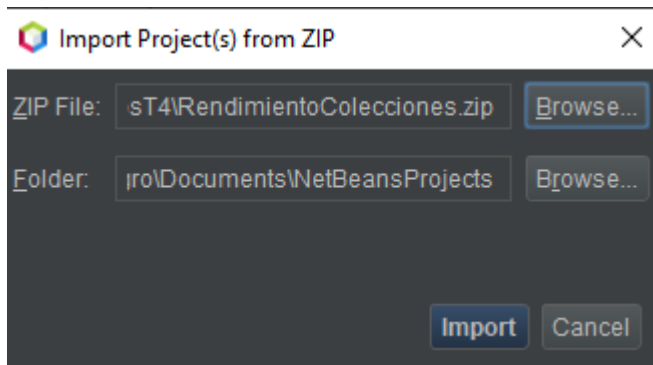
Tema 4: Optimización y Documentación

(Prácticas 4.4 y 4.11)

Tema 4: Optimización y Documentación	1
(Prácticas 4.4-4.11)	1
Práctica 4.4: Análisis de código Java en Netbeans:	2
Práctica 4.11: Practicar con la documentación de código java en Netbeans:	3

Práctica 4.4: Análisis de código Java en Netbeans:

- **Importa en Netbeans el proyecto RendimientoColecciones:**
Importamos el proyecto desde File > Import Project> from Zip:



Antes de empezar debemos solucionar un bug copiando el JDK16 de nuestra carpeta

profilerCorreccionBug\profilerCorreccionBug\lib\deployed\jdk16, y pegarla en ***netbeans\profiler\lib\deployed*** sustituyendo la actual.

- **Ejecuta el *profile* del mismo, configurando la sesión seleccionando “Métodos”.** Haz una captura de pantalla y comenta la diferencia de uso de CPU de los distintos métodos empleados.

Elegimos el main del proyecto, lo seleccionamos y le damos a Profile File. Una vez ahí en *Configure Session* seleccionamos métodos, y le damos a Profile:

Name	Total Time	Total Time (CPU)
main	1.498 ms (100 %)	1.233 ms (100 %)
rendimientoColecciones.Main.main (String[])	1.218 ms (81,3 %)	1.218 ms (98,8 %)
Self time	622 ms (41,5 %)	622 ms (50,5 %)
java.util.ArrayList.remove (int)	499 ms (33,4 %)	499 ms (40,5 %)
Self time	303 ms (20,2 %)	303 ms (24,6 %)
java.util.ArrayList.fastRemove (Object[], int)	196 ms (13,1 %)	196 ms (15,9 %)
java.lang.System.arraycopy[native] (Object, int, Object, int, int)	196 ms (13,1 %)	196 ms (15,9 %)
Self time	0,0 ms (0 %)	0,0 ms (0 %)
java.util.ArrayList.indexOf (Object)	80,6 ms (5,4 %)	80,6 ms (6,5 %)
java.util.ArrayList.indexOfRange (Object, int, int)	64,6 ms (4,3 %)	64,6 ms (5,2 %)
Self time	16,0 ms (1,1 %)	16,0 ms (1,3 %)
java.lang.Integer.valueOf (int)	15,9 ms (1,1 %)	15,9 ms (1,3 %)
org.netbeans.lib.profiler.server.ProfilerServer.activate (String, int, int, int)	265 ms (17,7 %)	0,0 ms (0 %)
org.netbeans.lib.profiler.server.ProfilerServer.doActivate (int)	265 ms (17,7 %)	0,0 ms (0 %)
org.netbeans.lib.profiler.server.ProfilerServer.delay (int)	265 ms (17,7 %)	0,0 ms (0 %)
java.lang.Thread.sleep[native] (long)	265 ms (17,7 %)	0,0 ms (0 %)
Self time	0,0 ms (0 %)	0,0 ms (0 %)
Self time	0,0 ms (0 %)	0,0 ms (0 %)
Self time	0,0 ms (0 %)	0,0 ms (0 %)
sun.launcher.LauncherHelper.checkAndLoadMain (boolean, int, String)	14,2 ms (0,9 %)	14,2 ms (1,2 %)
Reference Handler	0,0 ms (- %)	0,0 ms (- %)
Finalizer	0,0 ms (- %)	0,0 ms (- %)
Common-Cleaner	0,0 ms (- %)	0,0 ms (- %)
DestroyJavaVM	0,0 ms (- %)	0,0 ms (- %)

El ArrayList Remove() es el que más consume y el que menos ArrayList.indexOf().

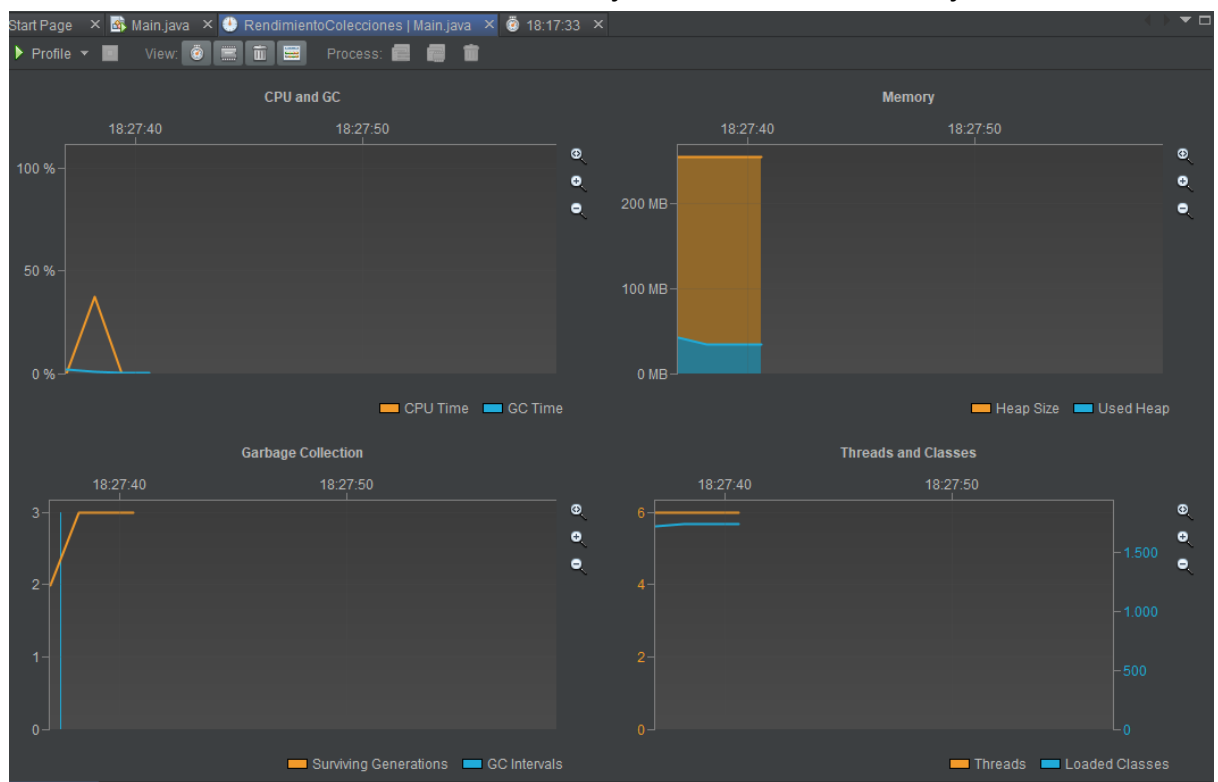
- Toma una *instantánea de análisis y guárdala*.

Para tomar la instantánea le damos a *Save Snapshot* arriba a la izquierda y nos aparecerá en snapshots. Necesitamos mostrar la ventana de snapshots, así que vamos a Window > Profiling > Snapshots.

Para guardar la instantánea hacemos click derecho en ella y le damos a Export seleccionando la ruta donde lo almacenemos.

- Ejecuta el *profile* opción *Telemetría* y haz una captura de pantalla.

Volvemos al Profiling y en vez de darle directamente a Profile seleccionamos la flechita a su derecha y le damos a Telemetry:

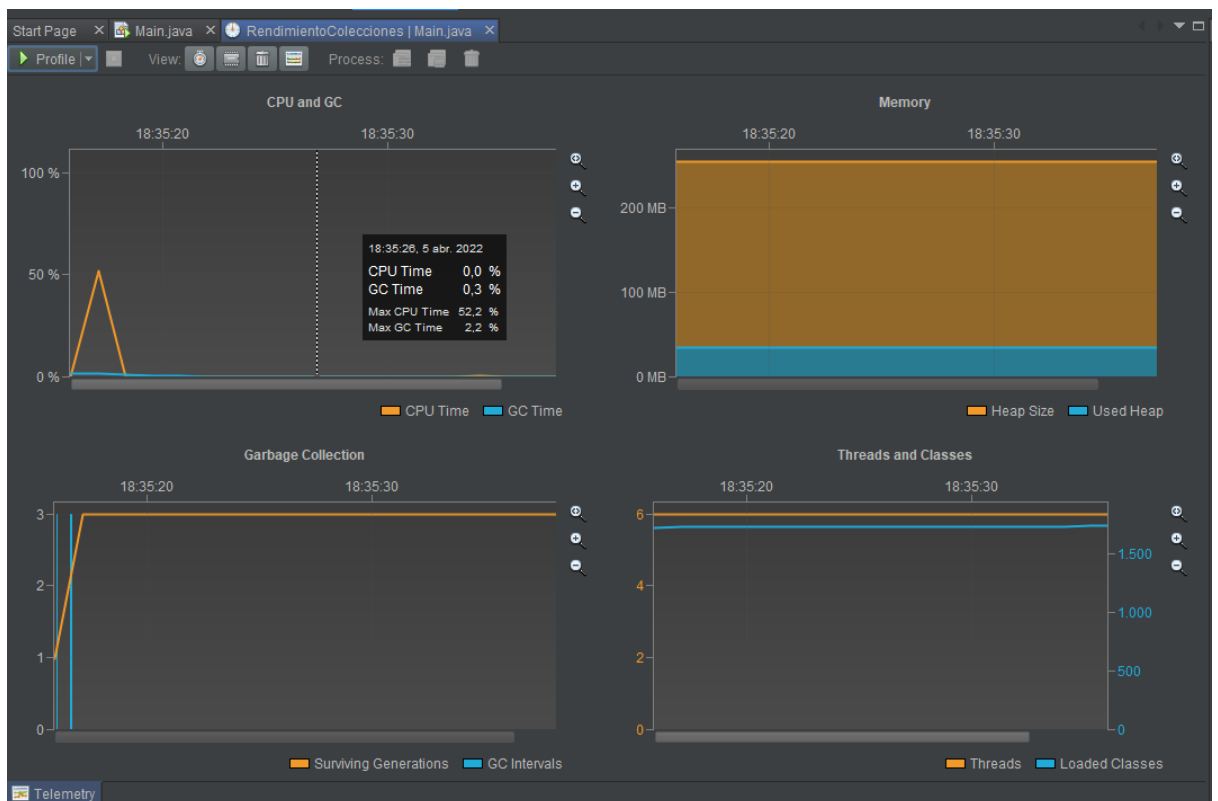


- Sustituye el ArrayList por LinkedList y repite todos los pasos anteriores.

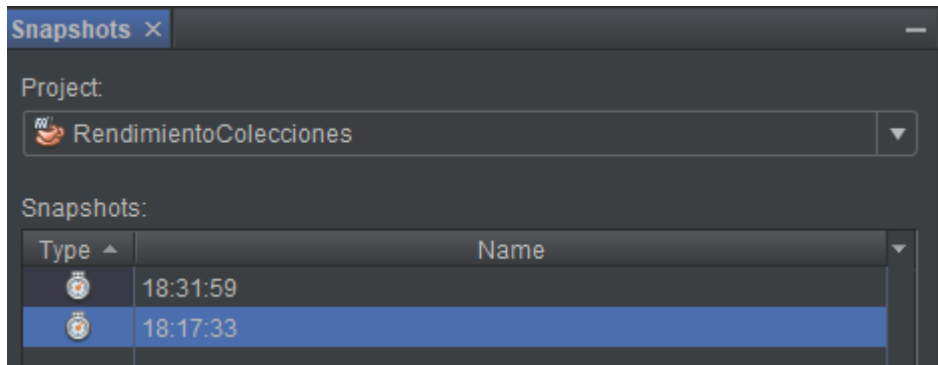
Name	Total Time	Total Time (CPU)
main	1.415 ms (100 %)	1.276 ms (100 %)
rendimientoColecciones.Main.main (String[])	1.276 ms (90,2 %)	1.276 ms (100 %)
Self time	601 ms (42,5 %)	601 ms (47,1 %)
java.util.ArrayList.remove (int)	576 ms (40,7 %)	576 ms (45,1 %)
Self time	398 ms (28,2 %)	398 ms (31,2 %)
java.util.ArrayList.fastRemove (Object[], int)	177 ms (12,5 %)	177 ms (13,9 %)
java.lang.System.arraycopy(native) (Object, int, Object, int, int)	177 ms (12,5 %)	177 ms (13,9 %)
Self time	0,0 ms (0 %)	0,0 ms (0 %)
java.util.ArrayList.indexOf (Object)	62,9 ms (4,4 %)	62,9 ms (4,9 %)
java.util.ArrayList.indexOfRange (Object, int, int)	62,9 ms (4,4 %)	62,9 ms (4,9 %)
Self time	0,0 ms (0 %)	0,0 ms (0 %)
java.lang.Integer.valueOf (int)	36,4 ms (2,6 %)	36,4 ms (2,9 %)
org.netbeans.lib.profiler.server.ProfilerServer.activate (String, int, int, int)	138 ms (9,8 %)	0,0 ms (0 %)
org.netbeans.lib.profiler.server.ProfilerServer.doActivate (int)	138 ms (9,8 %)	0,0 ms (0 %)
org.netbeans.lib.profiler.server.ProfilerServer.delay (int)	138 ms (9,8 %)	0,0 ms (0 %)
java.lang.Thread.sleep(native) (long)	138 ms (9,8 %)	0,0 ms (0 %)
Self time	0,0 ms (0 %)	0,0 ms (0 %)
Self time	0,0 ms (0 %)	0,0 ms (0 %)
Self time	0,0 ms (0 %)	0,0 ms (0 %)
Reference Handler	0,0 ms (- %)	0,0 ms (- %)
Finalizer	0,0 ms (- %)	0,0 ms (- %)
Common-Cleaner	0,0 ms (- %)	0,0 ms (- %)
DestroyJavaVM	0,0 ms (- %)	0,0 ms (- %)

El método remove sigue siendo el que más consume y el que menos también el mismo, indexOf.

Guardamos y exportamos la snapshot.



- **Haz una comparativa del rendimiento de LinkedList vs. ArrayList.**
Aunque consumen prácticamente lo mismo, LinkedList es un poco más óptimo debido a que consume un poco menos.
- **Haz una captura de la ventana SnapSHots en la que se vea la lista de snapshots hechos.**



Práctica 4.11: Practicar con la documentación de código java en Netbeans:

Documenta dos aplicaciones completas que tengas hechas (por ejemplo el proyecto Alumno de Refactorización o cualquiera de los proyectos de este o otros temas anteriores) hasta que “Analizar Javadoc” no de ningún aviso y por lo tanto consideremos que están totalmente documentadas.

Usa las etiquetas `@autor` y `@version` en las clases.

Muestra el resultado generado por Javadoc en formato HTML.

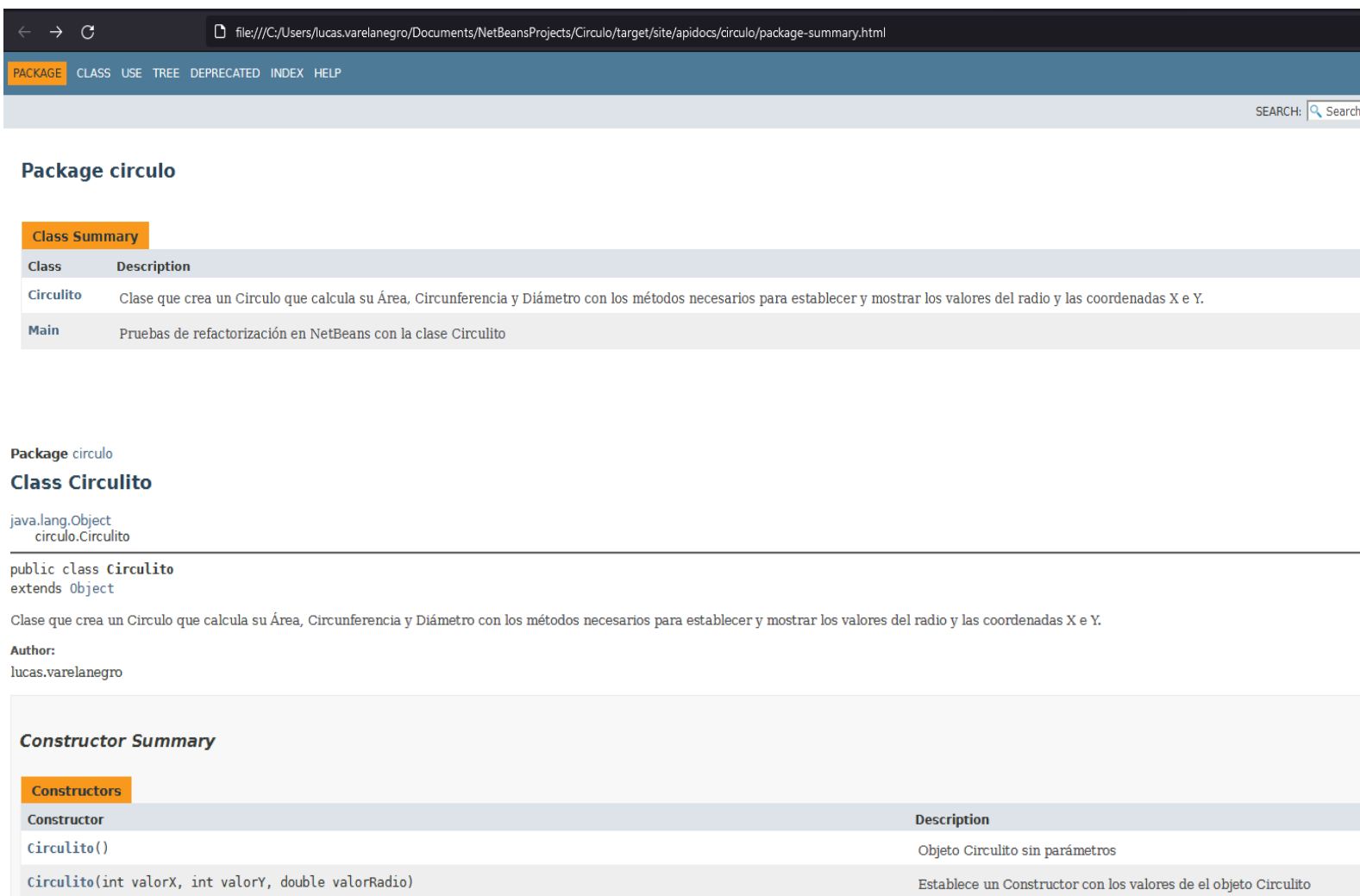
Proyecto Círculo:

El primer paso para realizar nuestra documentación será seleccionar el proyecto y en Tools hacer Analyze Javadoc.

Se nos abrirá una pestaña *Analyzer* con unas propuestas, las marcamos todas y le damos a Fix Selected.

Una vez hecho el Fix podremos documentar nuestro proyecto y al terminar iremos a Run y seleccionaremos Generate JavaDoc.

Resultado final:



file:///C:/Users/lucas.varelanegro/Documents/NetBeansProjects/Circulo/target/site/apidocs/circulo/package-summary.html

PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

SEARCH: Search

Package circulo

Class	Description
Circulito	Clase que crea un Círculo que calcula su Área, Circunferencia y Diámetro con los métodos necesarios para establecer y mostrar los valores del radio y las coordenadas X e Y.
Main	Pruebas de refactorización en NetBeans con la clase Circulito

Package circulo

Class Circulito

java.lang.Object
circulo.Circulito

```
public class Circulito
extends Object
```

Clase que crea un Círculo que calcula su Área, Circunferencia y Diámetro con los métodos necesarios para establecer y mostrar los valores del radio y las coordenadas X e Y.

Author:
lucas.varelanegro

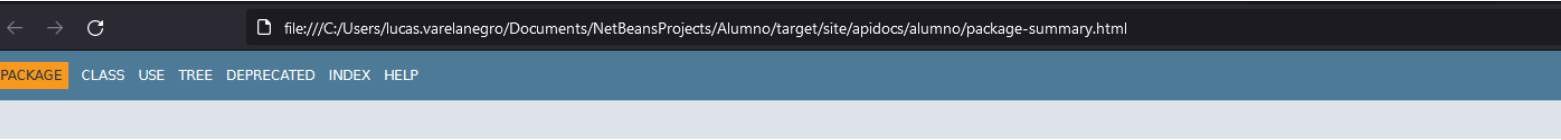
Constructor Summary

Constructors	Description
Circulito()	Objeto Circulito sin parámetros
Circulito(int valorX, int valorY, double valorRadio)	Establece un Constructor con los valores de el objeto Circulito

Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
int	getCoordenadaX()	Devuelve la CoordenadaX del Circulo
int	getCoordenadaY()	Devuelve la CoordenadaY del Circulo
double	getRadio()	Devuelve el valor del radio
double	ObtenerAreaCirculo()	Calcula y devuelve el Área del Circulo multiplicando PI por el radio elevado a 2
double	obtenerCircunferencia()	Calcula y devuelve la Circunferencia del Circulo multiplicando el diámetro del Circulo por PI
double	obtenerDiametro()	Devuelve el valor del diámetro del Circulo calculando el doble del radio
void	setCoordenadaX(int coordenadaX)	Establece un constructor de la CoordenadaX
void	setCoordenadaY(int coordenadaY)	Establece el constructor de la CoordenadaY
void	setRadio(double radio)	Establece el valor de el radio
String	toString()	Devuelve una cadena con las coordenadas X e Y y el radio del Circulo
void	trasladarCentro(int trasladarX, int trasladarY)	Método que desplaza las coordenadas de el Circulo estableciendo los nuevos valores de las coordenadas X e Y
Methods inherited from class java.lang.Object		
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait		

Proyecto Alumno:
 Realizamos los mismos pasos que en el anterior.
 Resultado final:



Package alumno

Class Summary	
Class	Description
Alumno	Clase hija de Persona que establece los parámetros de un Alumno, comprueba si es mayor de edad y establece y muestra su Curso y Teléfono
Main	Clase Main en el que se establecen 3 alumnos para probar el funcionamiento de las clases Persona y Alumno
Persona	Super clase que establece,comprueba y muestra los parámetros necesarios para una persona, nombre dni, fecha de nacimiento

Package alumno
Class Alumno
java.lang.Object
alumno.Persona
alumno.Alumno

public class Alumno
extends Persona
Clase hija de Persona que establece los parámetros de un Alumno, comprueba si es mayor de edad y establece y muestra su Curso y Teléfono
Author:
lucas.varelanegro

Constructor Summary

Constructors	
Constructor	Description
Alumno(String nombre, String dni, LocalDate fechaNacimiento)	Constructor de Alumno con su nombre, dni y fecha de nacimiento obtenidos de la clase Persona

Method Summary

All Methods	Instance Methods	Concrete Methods	
Modifier and Type		Method	Description
String		getCurso()	Método que devuelve el curso del alumno
long		getTelefono()	Método que devuelve el número de teléfono del alumno
void		setCurso(String curso)	Método que establece el curso del alumno
void		setTelefono(long telefono)	Método que establece el número de teléfono del alumno
Methods inherited from class alumno.Persona			
asignarLetraDNI, comprobarDNI, getDni, getFechaNacimiento, getNombre, setDni, setFechaNacimiento, setNombre, toString			
Methods inherited from class java.lang.Object			
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait			

Package alumno
Class Persona
java.lang.Object
alumno.Persona

Direct Known Subclasses:
Alumno
public class Persona
extends Object
Super clase que establece,comprueba y muestra los parámetros necesarios para una persona, nombre dni, fecha de nacimiento
Author:
lucas.varelanegro

Constructor Summary

Constructors	
Constructor	Description
Persona(String nombre, String dni, LocalDate fechaNacimiento)	Constructor para Persona con nombre, dni y fecha de nacimiento

Method Summary

All Methods	Instance Methods	Concrete Methods	
Modifier and Type		Method	Description
void		asignarLetraDNI()	Método que establece la letra del dni según su forma de calcularse
boolean		comprobarDNI()	Método que comprueba que el dni sea válido devolviendo un true en caso afirmativo y un false en caso contrario
String		getDni()	Método que devuelve el dni
LocalDate		getFechaNacimiento()	Método que devuelve la fecha de nacimiento
String		getNombre()	Método que devuelve el nombre
void		setDni(String dni)	Método que establece el dni
void		setFechaNacimiento(LocalDate fechaNacimiento)	Método que establece la fecha de nacimiento
void		setNombre(String nombre)	Método que establece el nombre
String		toString()	Método que devuelve el nombre, el dni y la fecha de nacimiento en una cadena
Methods inherited from class java.lang.Object			
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait			

