Control Nº4

**Threads y Exceptions**

**Nombre : Fecha: 27 de mayo 2016**

Considere lo siguiente: cada objeto debe encapsular sus atributos, generar getters y setters para cada uno de ellos. Crear un constructor por defecto que inicialice los atributos de la clase y un constructor que reciba como parámetro todos los atributos de la clase.

**Nombre del package cl.curso.java.control\_cuatro.jmaldonado**

Reemplazar jmaldonado por la inicial de su nombre y su apellido.

En los ejercicios prácticos se evaluará lo siguiente

* Uso de buenas practicas al escribir métodos, variables, clases e interfaces. (1 pts.)
* Identación del código. (1 pts.)
* Creación de Constructores (1 pts.)
* Encapsulación getters y setters (1pts)
* Uso de comentarios (1pts).

1. Indique Cuales son las formas de clasificar una Excepción. (1pts)

R: Formas de clasificar: Chequeadas, No chequeadas.

Chequeadas: Tienen como superclase a Exception, deben ser capturadas, de forma contraria el código lanza error y no compila.

No Chequeadas: Tienen como superclase a RuntimeException, No necesita ser capturada por medio del Try/Catch, pero si no se hace, la ejecución termina inesperadamente.

1. Las Excepciones que Heredan directamente de RuntimeException, ¿en que categoría entran? (1pts)

R: En Excepciones No Chequeadas.

1. ¿Cuales son las formas posibles para crear un Thread? de ejemplos. (1pts)

R:

\*Por medio de la implementacion de la clase Runnable.

public class EjemploRun implements Runnable {

public void run() {

System.out.println("Esto es un Hilo (no de costura)");

}

public static void main(String args[]) {

(new Thread(new EjemploRun())).start();

}

}

\*Por medio de extender la clase Thread.

public class EjemploHilo extends Thread {

public void run() {

System.out.println("Esto es un Hilo (Te dicen que costura NO)");

}

public static void main(String args[]) {

(new EjemploHilo()).start();

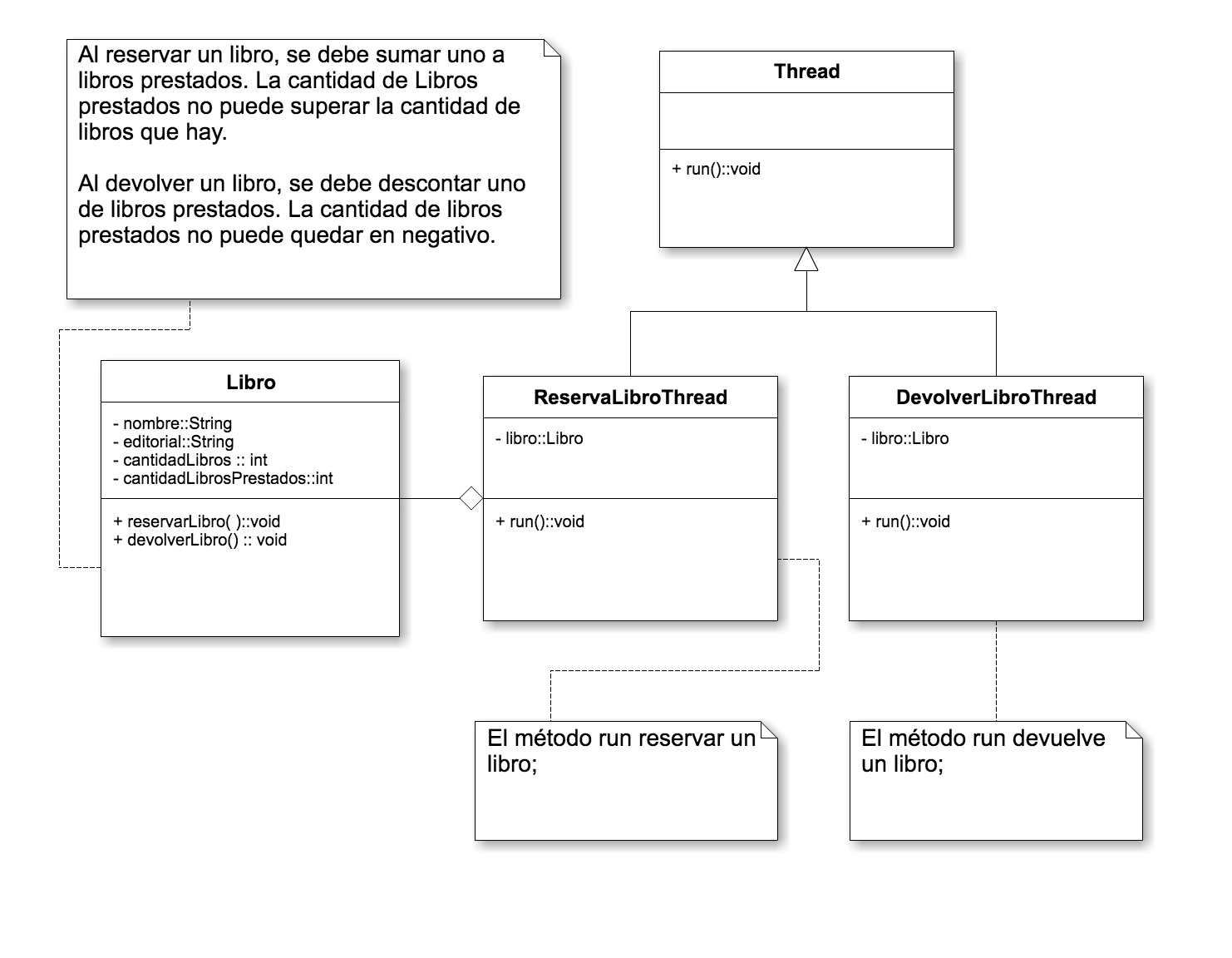
}

}

1. ¿Cual es el método para iniciar un thread en forma paralela ?

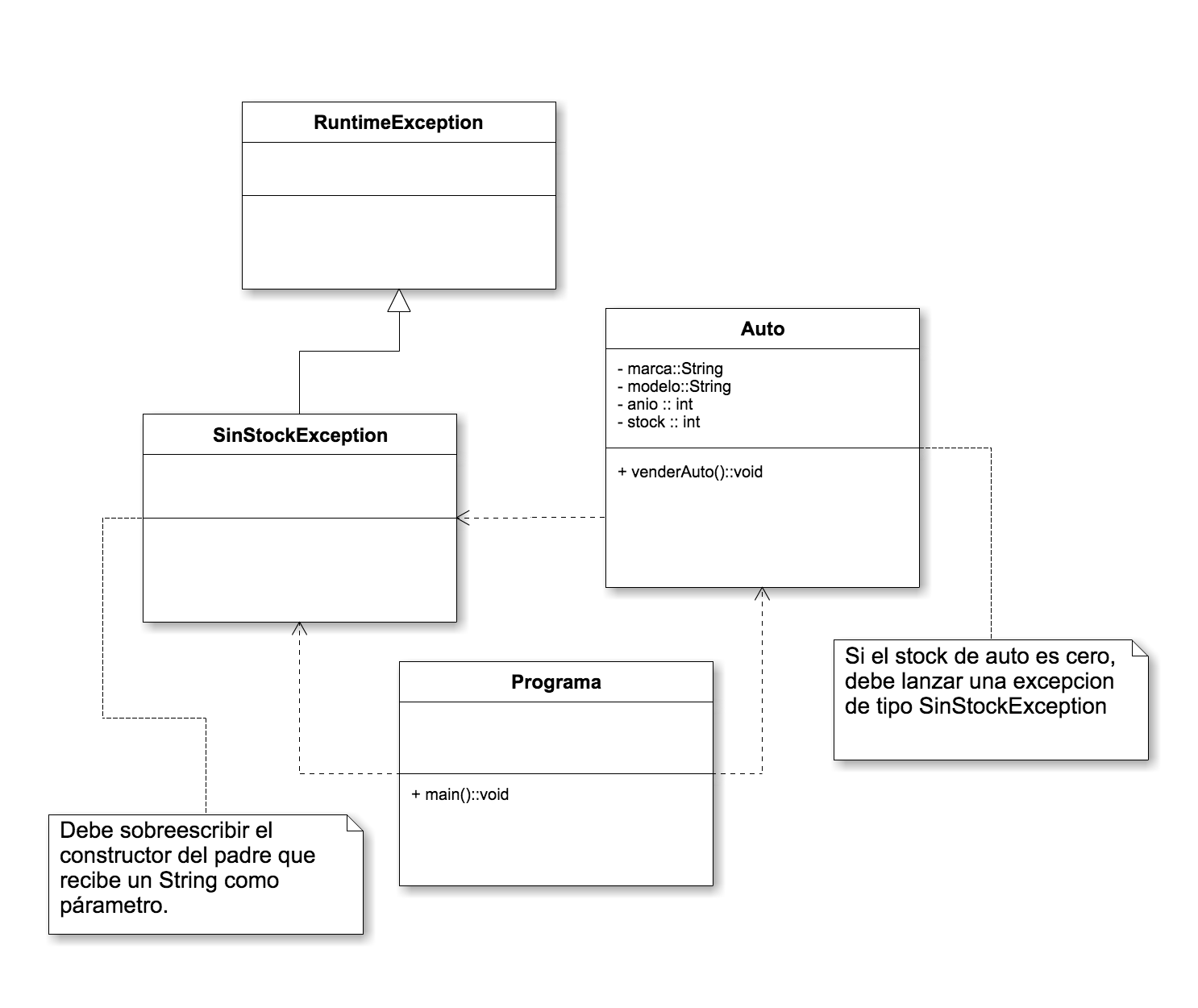
R: Metodo run() que debe ser sobreescrito, pero puede ser heredado de Thread, o implementado de Runnable.

1. Según el siguiente diagrama, escribir las clases necesarias.



* Escribir la clase Programa y crear una instancia de Libro , 3 instancias de ReservaLibro y 3 de DevolverLibro. Ejecutar los hilos. (1pts.)
* Debe sincronizar los métodos reservarLibro y devolverLibro. (1pts.)

1. Según el siguiente diagrama, escribir las clases necesarias.



* La clase programa debe crear una instancia de Auto y debe forzar a que ocurra la excepción SinStockException. El método main debe capturar dicha excepción e imprimir en consola “No es posible vender auto. No hay stock”. (1pts.)