Prueba Nº3

**Exception – Thread - ArrayList**

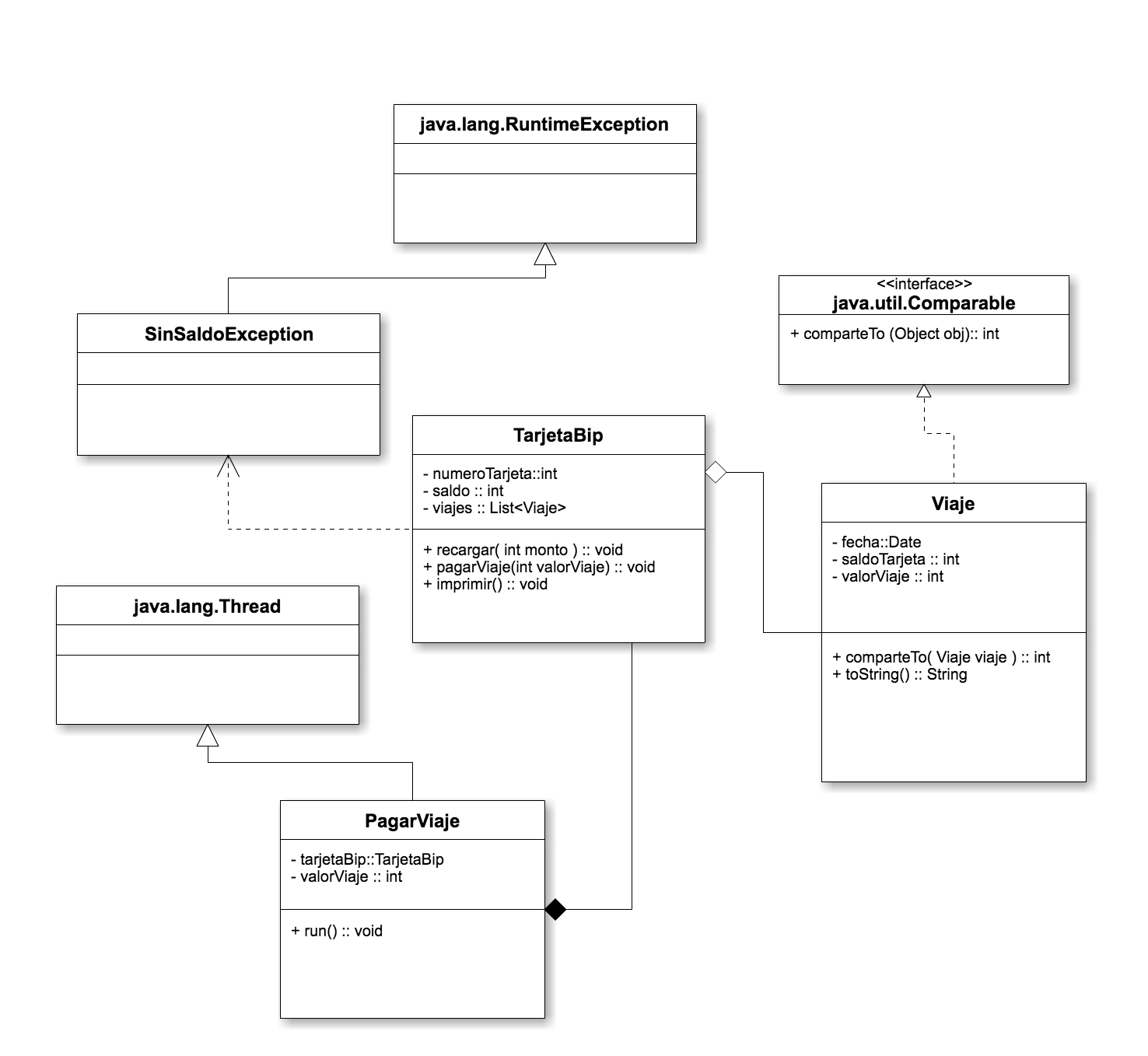
**Nombre : Fecha: 13 de junio 2016**

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.

* **Crear un proyecto maven para realizar la prueba. Group id: cl.curso.java.prueba\_tres; Artifact id: jmaldonado (1pto)**
  + Reemplazar jmaldonado por la *inicial* de su nombre y su apellido.

En los ejercicios prácticos se evaluará lo siguiente

1. Según el siguiente diagrama, escribir las clases necesarias. (16 ptos)



* El método TarjetaBip >> recargar( int monto ) , sumara el monto recibido por parámetro al saldo de la tarjeta bip. ( 1pts )
* El método TarjetaBip>>pagarViaje( int ValorViaje)::void descontara del saldo de la tarjeta según el valor del viaje recibido como parámetro (1pts).
  + Si el saldo de la tarjeta es menor al valor del viaje deberá lanzar la excepción SinSaldoException (1pts).
  + Si el saldo de la tarjeta es mayor o igual además de descontar del saldo de la tarjeta, se deberá agregar un nuevo Viaje a la lista de viajes de la tarjeta, al atributo fechaViaje se le asignara la fecha actual del sistema new Date() (1pto).
  + Este método debe ser sincronizado.
* El método TarjetaBip>>imprimir, deberá mostrar por consola el número de la tarjeta, el saldo y los viajes.
  + Debe ordenar la lista de viajes utilizando Collection.sort( viajes ) (1 pts.)
  + Debe recorrer la lista de viajes utilizando el ciclo foreach (1 pts.)
  + En cada iteración deberá imprimir cada viaje utilizando el método toString de la clase Viaje (1pts.)
* El método Viaje>>compareTo debe compara dos viajes utilizando el atributo fecha de la clase Viaje (1 pts.)
* El método Viaje>>toString debe retornar un String, con todos los atributos de la clase Viaje.(1pts)
* El método PagarViaje>>run deberá invocar al método pagar viaje de la tarjeta Bip (1pto)
* 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).

Opcional : Subir el código fuente de la prueba a un repositorio en github que se llame prueba-tres. Esto equivaldrá a un punto sobre la nota obtenida, es decir si su nota es un 3 subirá a un 4.