Lo primero es convertir todo lo que se da a java 8. En concreto:

1º Los fairgrounds de diferente tipo implementan el método dismantle, aunque la implementación se repite en casi todos, menos en el SuperFairGround. Soluciona esto usando Java 8

2º Se te entrega un Factory Method para que se puedan crear diferentes FairGround sin que importe cual sea el tipo. ¿Podrías convertirlo para usar el modelo que se propone en Java 8?

3º En la clase Statistic existen muchos recorridos a colecciones. ¿Podrías convertirlo a Java 8?

4º El BiasCostumer es un tipo de visitante al parque de atracciones. La cantidad máxima de atracciones a las que puede montarse viene dado en el propio constructor. Sin embargo el usuario puede que no complete ese número. Cada atracción que disfruta debe ser anotada por él con un valor real entre 0 y 10. Desde que se baja de una atracción hasta que se monta en la siguiente debe parar 1000 ms

La atracción en la que se va a montar es aleatoria, seguramente se repitan muchas, no importa. Cada vez que se baja de una atracción entregra una CustomerCard para valorar la atracción, de esta forma las atracciones podrán pasar a ser Super u otra vez, normales.

El usuario comienza con un valor real de positividad de 10 (currentJoyment) y tiene como cota mínima 5 (minimumEnjoyment) cada valoración modifica, calculando la media, el currentJoyment. Si dicho valor baja del minimumEnjoyment, el visitante termina la visita.

Al final entregará su currentJoyment para que, cuando termine el día de trabajo, se pueda saber el porcentaje de usuarios felices, en relación a la cantidad total de visitantes.

5º Te doy una clase que llamo WorkingDay, es invención pura, para facilitar la clase Park. Esta clase tiene:

* un objeto Park.
* Una lista de usuarios mediante la cual controla la cantidad de usuarios concurrentes, es decir, que están ahora mismo en el parque.
* Las propiedades necesarias para que pueda calcular, al final del día, el porcentaje de usuarios felices.

Este hilo va a terminar cuando se hayan alcanzado los 100.000 viajes (endOfTheDay), juntando todas las atracciones. Vamos a poner, por ejemplo, diez atracciones en el Park, da igual el tipo.

El WorkinDay también es el encargado de dar acceso (crear el hilo) a visitantes al parque. La cantidad de visitantes que va a admitir será la suficiente para que siempre tenga, mientras que no se haya llegado al endOfTheDay, cien visitantes concurrentes.

Al final del día también se debe mostrar la puntuación media de todas las atracciones juntas.

Q**ue se entrega**

Se te dan bastantes cosas:

En el paquete utils tienes Comment, una clase que permite encender o apagar los comentarios hechos con ella.

ObjetMother solo tiene el valor de minimumEnjoyment que debe cumplir el visitante para continuar su visita.

Las atracciones, ya conocidas y la arquitectura de FactoryMethod necesaria. El método dismantle() permite que una SuperFairGround vuelva a ser una FairGround normal, del tipo que sea.

La clase Fraction, muy útil para llevar las diferentes medías del programa, la valoración de la atracción, la valoración de la felicidad del visitante.

La clase FairGroundObservable es invención pura, se usa para separar la atracción de su valoración, es la que soporta el peso del observer. Como solo tenemos una clase pero hay dos posibles cambios, de normal a super o al revés, el destinatario de la observación, al que se notifica, la clase Statistics, debe preguntarse si dicho fairGroundObservable tiene el valor changeHigh en true. Si es así debe convertir el FairGround en SuperFairGround. En caso contrario sabrá que esta en una SuperFairGround y por tanto pasará a hacer un dismantle() para volver a convertirla en FairGround normal.

La clase Statistic es la que se encarga de las notas de las FairGrounds. Cuenta con la lista de FairGrounds, que se obtiene del parámetro Park del constructor. changeRate añade la nota a la FairGrounds pasada por parámetro. Cuando hace el update, que es llamado desde el Observable, modifica la lista del park pero además también modifica el mapa que asocia cada FairGround con su clase de puntuacion FairGroundsObservable.

Finalmente tiene un método getCurrentAverageRate. Que calcula la nota media de todas las atracciones. Un aviso, este cálculo no puede hacerse hasta que no haya 0 (cero) visitantes activos. Me explico: EndOfTheDay sirve para no aceptar (crear) más visitantes, pero sigue habiendo visitantes hasta que no terminen su visita