Realidad Distorsionada de Lovecraft

Índice de Clases:

Public class Atracción implements Comparable<Atracción>

En la clase atracción se guardan los datos de las atracciones: string *nombre*, double *costo*, double *tiempo promedio*, int *cupo diario* y string *tipo*. Además, la misma implementa Comparable para poder ordenarse con otra atracción.

Métodos:

- public String getNombre()

Obtiene el nombre de Atracción.

- public String getTipo()

Obtiene el tipo de Atracción.

- public double getCosto()

Obtiene el costo de Atracción.

- public double getTiempoPromedio()

Obtiene el tiempo de Atracción.

- public int getCupoDiario()

Obtiene el cupo diario de Atracción.

- public boolean tieneCupo()

Revisa si la Atracción tiene cupo.

- public void comprar()

Reduce el cupo diario de la Atracción.

- public String toString()

Devuelve un String con: el nombre de la Atracción, el precio y la duración en horas.

- public int hashCode()

Devuelve un hash generado con: el costo, el cupo diario, el nombre, el tiempo promedio y el tipo.

- public boolean equals(Object obj)

Compara 2 Atracciones.

- public int compareTo(Atracción otra)

Compara 2 Atracciones por costo, por tiempo, por nombre y por tipo, respectivamente. Las ordena de mayor a menor.

Public class Promocion implements Comparable<Promoción>

En la clase Promoción se guardan los datos de las promociones: protected String *nombre*, protected String *tipo de promoción*, protected int *cantidad de atraccione*s que posee, protected double *duración*(Tiempo total a invertir, la suma del tiempo dé las Atracciones que se incluyen en la Promoción), protected double *precio original*(Precio de la promoción sin descuento), protected double *precio a mostrar*(Precio con el descuento correspondiente aplicado que se mostrará al usuario). Estos datos se guardan como Protected así las clases hijas que corresponden a los 3 tipos de promoción, pueden heredarlos. Además, la clase implementa Comparable para poder ordenarse con otra Promoción.

Métodos:

- public Promoción(String nombre, int cant_atracciones, Atraccion[] atracciones, String tipo_promocion)

Constructor de Promoción.

- protected String obtenerAtraccionesIncluidas()

Obtiene las Atracciones que se incluyen en la Promoción.

- public double getDuracion()

Obtiene la duración de la Promoción.

- public double getPrecio_mostrar()

Obtiene el precio que se le mostrará al usuario.

- public int compareTo(Promoción otra)

Compara 2 Promociones por costo, por tiempo, por nombre y por tipo, respectivamente. Las ordena de mayor a menor.

- public boolean tieneCupo()

Verifica si las atracciones que están incluidas dentro dé la Promoción tienen cupo.

- public String[] obtenerNombresAtracciones()

Obtiene el nombre de las Atracciones que se incluyen en la Promoción.

- public void comprar()

Reduce el cupo de las atracciones que incluye la promoción.

- public String toString()

Devuelve un String con: el nombre de la promoción, el tipo de la promoción, la duración en horas, el precio original y las atracciones incluidas.

Public class PromocionAbsoluta extends Promoción

Esta clase se extiende de Promoción para poder hacer uso tanto de sus atributos como sus métodos. La Promoción absoluta establece un precio específico por las atracciones que incluye.

Métodos:

- public PromocionAbsoluta(String nombre, int cant_atracciones, Atraccion [] atracciones, String tipo_promocion, double precio_absoluto)

Constructor de Promoción Absoluta (Utiliza la clase Promoción).

- public String toString()

Devuelve un String con: el toString del Padre, el precio original, el descuento aplicado y el precio final de la promoción.

Public class PromocionCombo extends Promoción

En la clase PromocionCombo se guarda el dato: private String *nombre de la atracción gratuita*(Propio del tipo de Promoción). Esta clase se extiende de Promoción para poder hacer uso tanto de sus atributos como sus métodos.

Métodos:

- public PromocionCombo(String nombre, Atraccion[] atracciones, String tipo_promocion, Atraccion atraccion_gratis)

Constructor de Promoción Combo (Utiliza la clase Promoción).

- public String getAtraccionGratis()

Obtiene la Atracción gratuita que incluye la Promoción.

- public boolean tieneCupo()

Verifica si la Atracción gratuita incluida tiene cupo.

- public void comprar()

Llama al comprar del Padre y reduce el cupo de la atracción gratuita que incluye la promoción.

- public String toString()

Devuelve un String con: el toString del Padre, la atracción gratuita, la cantidad de dinero ahorrada y el precio final de la promoción.

Public class PromocionPorcentual extends Promoción

En la clase PromocionPorcentual se guarda el dato: private double *descuento*(Cantidad de dinero ahorrado) y private double *porcentaje*(Porcentaje de Descuento, propio del tipo de promoción). Esta clase se extiende de Promoción para poder hacer uso tanto de sus atributos como sus métodos.

Métodos:

- public PromocionPorcentual(String nombre, int cant_atracciones, Atraccion [] atracciones, String tipo_promocion, double porcentaje)

Constructor de Promoción Porcentual (Utiliza la clase Promoción).

- public double getDescuento()

Obtiene el descuento de la Promoción Porcentual.

- public String toString()

Devuelve un String con: el toString del Padre, el porcentaje de descuento, la cantidad de dinero ahorrada y el precio final de la promoción.

Public class Usuario

En la clase Usuario se guardan los datos: private String *nombre*, private String *preferencia*(Tipo de atracción que prefiere el usuario), private int *dinero* y private double *tiempo*. Además, se utilizan:

- ☐ Un HashSet, donde se guardan las atracciones que son aceptadas por el usuario, que será utilizado para verificar si un Usuario ya aceptó esta Atracción o no.

- ☐ 2 ArrayList, para guardar una lista de promociones y una lista de atracciones que se mostrarán en el archivo de salida.

Métodos:

- public Usuario(String nombre, String preferencia, int dinero, double tiempo)

Constructor de Usuario.

- public String getPreferencia()

Obtiene la Preferencia del Usuario.

- public String getNombre()

Obtiene el Nombre del Usuario.

- public String salidaUsuario()

Retorna un String con la información para el archivo de salida de cada usuario..

- public String toString()

Devuelve un String con: el nombre del Usuario, su preferencia, su dinero y su tiempo.

- public boolean puedeComprarPromocion(Promocion promo)

Verifica si el Usuario puede comprar una Promoción, revisa si posee el tiempo y dinero suficiente, si las Atracciones de la Promoción tienen cupo y si el Usuario no acepto ya una Atracción que pertenezca a la Promoción.

- private boolean tieneAceptada(Promocion promo)

Verifica si el usuario ya aceptó alguna Atracción que pertenece a la Promoción.

- public void agregarPromocion(Promocion promo)

Añade la Promoción a la lista de promociones aceptadas y ,además, añade las atracciones de la Promoción al HashSet de atracciones aceptadas.

- public boolean puedeComprarAtraccion(Atraccion atraccion)

Verifica si el Usuario puede comprar una Atracción, revisa si posee el tiempo y dinero suficiente, si la Atracción tiene cupo y si el Usuario no acepto ya esa Atracción.

- private boolean tieneAceptada(Atraccion atraccion)

Verifica si el usuario ya aceptó esa Atracción.

- public void agregarAtraccion(Atraccion atraccion)

Añade la Atracción a la lista de atracciones aceptadas y al HashSet de atracciones aceptadas.

Public abstract class Archivo

La clase Archivo se dedica a la lectura de archivos.

Métodos:

- public void leerArchivo(String ruta, Datos carga)

Se encarga de la lectura del Archivo.

- protected abstract void cargarFormato(int cant, Datos carga, Scanner scanner)

Método cargado para la utilización por las clases hijas.

Public class ArchivoAtracciones extends Archivo

La clase ArchivoAtracciones se dedica a la lectura de los archivos de Atracciones.

Métodos:

- protected void cargarFormato(int cant, Datos carga, Scanner scanner)

Lee el archivo de Atracciones y genera las Atracciones que serán guardadas en un HashMap de atracciones y en otro HashMap de atracciones por tipo que apunta a un TreeSet de Atracciones.

- private void cargarMapaTipos(HashMap<String, TreeSet<Atraccion>> mapa, Atraccion cargada, String tipo)

Carga el tipo de Atracción al HashMap (si no la contiene) y luego añade la Atracción.

Public class ArchivoPromociones extends Archivo

La clase ArchivoPromociones se dedica a la lectura de los archivos de Promociones.

Métodos:

- protected void cargarFormato(int cant, Datos carga, Scanner scanner)

Lee el archivo de Promociones y genera las Promociones que serán guardadas en un HashMap de promociones por tipo.

- private void cargarMapaTipos(HashMap<String, TreeSet<Promocion>> mapa, Promocion cargada, String tipo, String tipo_promocion)

Carga el tipo de Atracción al HashMap (si no la contiene) y luego añade la Promoción.

- private Promocion generarPromocion(String nombre_promocion, String preferencia, int cant_atracciones, String tipo, Atraccion[] atracciones, String random, Datos carga)

Genera la Promoción dé acuerdo al tipo de promoción(Porcentual, Absoluta o Gratis).

Public class ArchivoSalida

La clase ArchivoSalida se encarga de la generación del archivo de salida y guarda la *ruta* donde se generará.

Métodos:

- public void escribirSalida(Datos cargados)

Escribe el archivo de salida utilizando el método: informeUsuarios.

- private void informeUsuarios(LinkedList<Usuario> lista_usuarios, PrintWriter escritor)

Escribe el archivo de salida con los datos correspondientes por cada usuario.

Public class ArchivoUsuarios extends Archivo

La clase ArchivoUsuarios se dedica a la lectura de los archivos de Usuarios.

Métodos:

- protected void cargarFormato(int cant, Datos carga, Scanner scanner)

Lee el archivo de Usuarios y genera los Usuarios que serán guardados en una LinkedList de usuarios.

Public class Datos

La clase Datos se encarga de todo el manejo de los mismos. Utiliza las listas y mapas generados por clases de Archivo(Atracción,Promoción y Usuarios). Por otro lado, se generan 2 HashMaps que apuntan a TreeSets(uno a Promoción y otro a Atracción) para guardar las Atracciones y Promociones no preferidas por el usuario(Que no corresponden a la Preferencia elegida).

Métodos:

- public LinkedList<Usuario> getLista_usuarios()

Obtiene la Lista de Usuarios.

- public void setLista_usuarios(LinkedList<Usuario> lista_usuarios)

Genera la Lista de Usuarios.

- public HashMap<String, Atraccion> getMapa_atracciones()

Obtiene el Mapa de Atracciones.

- public void setMapa_atracciones(HashMap<String, Atraccion> mapa_atracciones)

Genera el Mapa de Atracciones.

- public HashMap<String, TreeSet<Atraccion>> getMapa_atracciones_tipos()

Obtiene el Mapa de Atracciones por Tipo.

- public void setMapa_atracciones_tipos(HashMap<String, TreeSet<Atraccion>> mapa_atracciones_tipos)

Genera el Mapa de Atracciones por Tipo.

- public String obtenerTipoAtraccion(String nombre_atraccion)

Retorna el tipo de Atracción revisando el Mapa de Atracciones.

- public HashMap<String, TreeSet<Promocion>> getMapa_promos_tipos()

Obtiene el Mapa de Promociones por Tipo.

- public void setMapa_promos_tipos(HashMap<String, TreeSet<Promocion>> mapa_promos_tipos)

Genera el Mapa de Promociones por Tipo.

- public HashMap<String, TreeSet<Promocion>> getMapa_no_preferencia_promociones()

Obtiene el Mapa de las Promociones no preferidas por el Usuario.

- public HashMap<String, TreeSet<Atraccion>> getMapa_no_preferencia_atraccion()

Obtiene el Mapa de las Atracciones no Preferidas por el Usuario.

- public void ordenarNoPreferencias()

Ordena los Mapas de las Atracciones y Promociones no preferidas por el Usuario.

- private TreeSet<Promocion> calcularArbolTotalPromocion()

Genera un árbol con todas las Promociones sin tener en cuenta los tipos.

- private TreeSet<Atracción> calcularArbolTotalAtraccion()

Genera un árbol con todas las Atracciones sin tener en cuenta los tipos.

- private TreeSet<Promocion> obtenerArbolPromocionSinTipo(String descartar_tipo)

Genera el árbol de las Promociones cuyo Tipo no es el preferido por el usuario.

- private TreeSet<Atraccion> obtenerArbolAtraccionSinTipo(String descartar_tipo)

Genera el árbol de las Atracciones cuyo Tipo no es el preferido por el usuario.

- public Atraccion obtenerAtraccion(String nombre_atraccion)

Devuelve la Atracción guardada en el Mapa de Atracciones.

Public class Lector

La clase Lector se dedica a la lectura de datos y guarda la ruta donde se deben buscar los distintos archivos(usuarios.in, atracciones.in y promociones.in).

Métodos:

- public Datos leerDatos()

Genera los distintos archivos(ArchivoUsuarios, ArchivoPromociones y ArchivoAtracciones) y los datos. Luego lee los archivos, carga los Datos y los ordena para finalmente devolverlos.

Public class Menu

La clase Menu muestra la interfaz con la que interactúa el Usuario

Métodos:

- public void mostrar(Datos carga)

Muestra por pantalla la interfaz con la que interactúa el usuario.

- private void ofrecerPromociones(Usuario usuario_actual, TreeSet<Promocion> arbol, Scanner scanner)

Ofrece la Promoción al usuario si el mismo puede comprarla y la agrega a su lista en caso afirmativo.

- private void ofrecerAtracciones(Usuario usuario_actual, TreeSet<Atraccion> arbol, Scanner scanner)

Ofrece la Atracción al usuario si el mismo puede comprarla y la agrega a su lista en caso afirmativo.

- private boolean mostrarPromocion(Promocion promo, Scanner scanner)

Permite al usuario aceptar o rechazar una Promoción.

- private boolean mostrarAtraccion(Atraccion atrac, Scanner scanner)

Permite al usuario aceptar o rechazar una Atracción.