

1.- Clase InOut:

Clase que permite la lectura y escritura de ficheros con el objetivo de convertir y almacenar las preguntas en formato csv a formato Pregunta (clase), con la que trabajarán otras clases del proyecto.

Métodos:

importarPreguntas(), addPreg(String, HashSet<Pregunta>, HashSet<String>),
leerPreguntasDelSistema(), guardarPreguntasEnElSistema(HashSet<Pregunta>),
agrandarBateriaPreguntas(HashSet<Pregunta>, HashSet<Pregunta>),
importarPreguntasPath(String) (DEPRECATED).

2.- public static HashSet<Pregunta> importarPreguntas():

Método estático que abre un selector de archivos con el fin de que el usuario escoja un fichero .csv con preguntas para importar. Se apoya en el método addPreg(...) para convertir las preguntas a tipo Pregunta.

Devuelve un HashSet de tipo Pregunta con las preguntas importadas.

3.- static public void addPreg(String s, HashSet<Pregunta> hs1, HashSet<String> hs2):

Método estático que convierte un String (s) en una instancia de pregunta, y la mete en el HashSet hs1. Si no puede ser convertido en instancia de pregunta (está mal), lo mete en hs2.

No devuelve nada ya que las preguntas se añaden desde el propio método

4.- public static HashSet<Pregunta> importarPreguntasPath(String):

Método estático obsoleto. Se utiliza para debug, tiene la misma funcionalidad que importarPreguntas() con la salvedad que en vez de abrir un FileChooser, se le pasa el path del archivo como parámetro (de tipo String).

5.- public static HashSet<Pregunta> agrandarBateriaPreguntas(HashSet<Pregunta>, HashSet<Pregunta>):

Método estático simple, que devuelve un HashSet tipo Pregunta a partir de sumar los HashSet<Pregunta> pasados como parámetro.

6.- public static void guardarPreguntasEnElSistema(HashSet<Pregunta>):

Método estático que escribe un fichero tipo objetos en el directorio padre a la jerarquía de archivos a partir de un HashSet de objetos tipo Pregunta pasado como parámetro.

7.- public static HashSet<Pregunta> leerPreguntasDelSistema()

Método que lee el fichero escrito por guardarPreguntasEnElSistema(HashSet<Pregunta>).

Devuelve un HashSet con instancias de Pregunta.

8.- Clase Examen:

Abstracción de un examen tipo test.

Atributos: private String nombre, private int nPreguntas, private BancoPreguntas supply, private ArrayList<Pregunta> banco, private int rateFalladas, private BancoFalladas bancoFalladas, private HashSet<Pregunta> preguntasHS, private ArrayList<Pregunta> preguntasExamen.

Métodos: getPreguntasExamenHS(), getPreguntasExamen(), getRate(int, int), generarExamen(int, int), getSize(), getCorrectas().

9.- public ArrayList<String> getCorrectas():

Método que devuelve un ArrayList de Strings, cada uno de estos valores siendo la respuesta correcta para cada una de las preguntas del examen (letra de la A a la D).

10.- public ArrayList<Pregunta> generarExamen(int npreguntas, int ratefalladas):

Método que devuelve un ArrayList con las preguntas del examen a partir de los parámetros pasados npreguntas (número de preguntas deseado) y ratefalladas (el porcentaje de preguntas que han sido falladas en otros exámenes que se quieren dentro de este).

Esto se hace a partir de los banco de preguntas que tenga la instancia (bancoPreguntas y bancoFalladas), mediante el método random() de la clase Math.

Actualmente, la funcionalidad de añadir preguntas que han sido falladas en otros exámenes no está implementado en el resto de la aplicación, por lo que siempre se le pasará un 0 a este método.