



Ejercicio Nivel 8

InscripciónCedulas

Descripción

Se necesita una aplicación para manejar la información de los deportes más populares en Colombia y los datos de los deportistas nacionales más sobresalientes. La aplicación debe permitir visualizar los deportes registrados y sus deportistas correspondientes.

Objetivos

El objetivo de este ejercicio es que el estudiante comprenda y adquiera práctica en:

- El desarrollo de aplicaciones siguiendo un proceso incremental.
- La creación y captura de distintos tipos de excepciones para informar al usuario de manera conveniente cualquier problema detectado.
- La construcción de métodos para hacer persistir la información del modelo del mundo por medio del mecanismo de serialización.
- La construcción de métodos para importar y exportar información desde y hacia archivos secuenciales de texto, utilizando las clases que provee Java para manejo de archivos y manipulación de cadenas de caracteres.
- La utilización del elemento gráfico JComboBox de Java como mecanismo para manejar listas desplegables en la interfaz de usuario.

Los siguientes pasos conforman el plan sugerido para desarrollar el ejercicio. La idea es ir desarrollando y probando incrementalmente los métodos de las clases.

Este ejercicio debe ser realizado de manera **INDIVIDUAL**.

Preparación

1. Ejecute del sitio web del curso el archivo demo de la aplicación (**n8_inscripcionCedulas_demo.mp4**) y ejecútelo para conocer el funcionamiento esperado del programa.
2. Descargue del sitio web del curso el esqueleto del ejercicio (del enlace llamado **n8_inscripcionCedulas_esqueleto**) del ejercicio del sitio del curso. Descomprima este archivo e importe el proyecto llamado **n8_inscripcionCedulas** en Eclipse.
3. Lea el enunciado del problema disponible en:
n8_inscripcionCedulas/docs/specs/Descripcion.pdf.

4. Estudie el documento de requerimientos funcionales disponible en:
n8_inscripcionCedulas/docs/specs/RequerimientosFuncionales.pdf.
5. Estudie el documento de requerimientos no funcionales disponible en:
n8_inscripcionCedulas/docs/specs/RequerimientosNoFuncionales.pdf.
6. Estudie el documento de consideraciones adicionales de diseño disponible en:
n8_inscripcionCedulas/docs/specs/ConsideracionesAdicionalesDeDisenno.pdf.
7. Estudie el modelo del mundo diseñado para este ejercicio. Este modelo se encuentra en:
n8_inscripcionCedulas/docs/specs/ModeloConceptual.jpg. Identifique las clases, relaciones entre clases, constantes, atributos y métodos
8. Asegúrese de tener activado el uso de aserciones para la ejecución del programa. Ver el tutorial en:
http://cupi2.uniandes.edu.co/sitio/images/cursosCupi2/apo2/tutoriales/n7_assert.pdf

Parte1: Construcción y prueba de invariantes

1. Complete la clase **Ciudadano**
 - a. Completar los TODOs que comienzan por: **//TODO Parte1 PuntoY**
 - b. Utilice (invoque) el método `verificarInvariante()` para verificar el invariante en todos aquellos métodos de la clase que modifican el estado.
2. Complete la clase **PuestoVotacion**
 - a. Completar los TODOs que comienzan por: **//TODO Parte1 PuntoY**
 - b. Utilice (invoque) el método `verificarInvariante()` para verificar el invariante en todos aquellos métodos de la clase que modifican el estado.

Parte2: Creación de nuevas excepciones

1. Revise en el documento de descripción del problema, los tipos de errores que se pueden presentar en la ejecución del programa y la acción que se debe realizar en cada caso.
2. Revise el modelo del mundo e identifique las clases que permiten manejar las excepciones de la aplicación. Estudie su relación y dependencia con el resto del modelo.
3. Implemente y documente las clases propias de excepción de la aplicación. Estas son: **PersistenciaException**, **FormatoArchivoException** y **ElementoExisteException**. Para su implementación, puede guiarse por su definición en el modelo conceptual y en el documento de consideraciones adicionales de diseño.

Parte 3: Implementación de persistencia por serialización-deserialización

Complete los TODOs que comienzan por: **//TODO Parte3 PuntoY**

Parte 4: Uso de la excepción ElementoExisteException

Completar los TODOs que comienzan por: **//TODO Parte4 PuntoY**

Parte 5: Importación y exportación de información (lectura y escritura de archivos secuenciales de texto)

Completar los TODOs que comienzan por: **//TODO Parte5 PuntoY**

Parte 6: Creación de listas desplegables en la interfaz

Completar los TODOs que comienzan por: **//TODO Parte6 PuntoY**

Validación

Para comprobar el funcionamiento de su ejercicio usted puede:

1. Ejecutar las pruebas automáticas disponibles en el ejercicio (validación funcional del mundo) que le permitirán verificar la correcta implementación de sus métodos. Para ejecutar las pruebas en el entorno de Eclipse presione click derecho sobre el paquete “uniandes.cupi2.observatorio.test” (o alguna de sus clases internas), seleccione la opción *Run as* y posteriormente la opción *JUnitTest*. En el siguiente vínculo encuentra un video que le explica cómo ejecutar e interpretar las pruebas automáticas:

<https://www.youtube.com/watch?v=h3r7wSFaIOo>

2. Ejecutar el programa e interactuar con todas las opciones disponibles en la interfaz.

Entrega

Este ejercicio debe ser realizado de manera **INDIVIDUAL**.

1. Construya el archivo entregable con el ejercicio completo. No olvide revisar que su entrega cumple con lo especificado en las normas del curso referentes a entregas de ejercicios. Consultar:

<http://cupi2.uniandes.edu.co/sitio/index.php/cursos/apo1/normas-administrativas>

2. Entregue el archivo del ejercicio vía SicuaPlus, de acuerdo con las normas, fecha y hora de entrega.

NOTA. No olvide:

1. Renombrar el archivo de entrega con su login de estudiante. El nombre del archivo que contiene el ejercicio debe cumplir el siguiente formato:

n<nivel del ejercicio>_<login estudiante>.zip
(por ejemplo: n8_tsuarez.zip)

2. En el siguiente vínculo encuentra un video que le explica cómo indentar el código de su proyecto:

<https://www.youtube.com/watch?v=BH9H0e-Z56E>