UNIVERSIDAD ICESI

Algoritmos y Programación II
Miniproyecto 1

Objetivos de unidad

Capítulo 1: Búsqueda, ordenamiento y pruebas automáticas

- Escribir el invariante de una clase e implementar los métodos necesarios para su verificación, utilizando la instrucción assert de Java.
- Desarrollar las clases y los métodos necesarios para implementar las pruebas unitarias automáticas, que ayudan a comprobar el correcto funcionamiento de un programa.
- Entender la importancia de construir programas correctos y la manera como los invariantes, los contratos de los métodos y las pruebas unitarias son fundamentales en este propósito.
- Implementar una interfaz de usuario en la cual se despliegue y se permita la interacción con un grupo de objetos, utilizando algunos de los componentes gráficos básicos que ofrece el lenguaje Java.
- Implementar, adaptar y utilizar algunos algoritmos clásicos de búsqueda de información en una estructura ordenada o desordenada. Implementar, adaptar y utilizar algunos de los algoritmos no recursivos de ordenamiento de información.

Capítulo 2: Archivos, serialización y tipos de excepción

- Hacer persistir el estado del modelo del mundo del problema al terminar la ejecución de un programa y restaurarlo al volver a ejecutar el mismo.
- Manipular archivos de texto y utilizarlos para implementar algunos requerimientos del cliente.
- Usar e implementar distintos tipos de excepción como parte de un programa, de manera que sea posible clasificar los tipos de error que se pueden presentar y asociarles en el programa distintas maneras de recuperarse ante el problema.
- Construir las pruebas unitarias automáticas para el caso de manejo de archivos, persistencia y excepciones, y utilizarlas como un mecanismo de construcción de programas correctos de manera incremental.
- Utilizar el depurador de Eclipse como una ayuda adicional en el proceso de desarrollo de programas.

CONDICIONES DEL EJERCICIO

1. Para este ejercicio usted debe realizar el proceso completo de desarrollo de software. Lo que significa que su carpeta de entrega del ejercicio debe tener adentro 4 subcarpetas así:

    - **Planeación del proyecto:**
        1. Cronograma de trabajo y roles
    - **Análisis:**
        1. Documento de requerimientos funcionales y no funcionales con formato Cupi2.
    - **Diseño:**
        1. Diagrama de clases del mundo, de la interfaz y de las pruebas.
        2. Diseño de pantallas y Mapa de navegación

    - **Implementación:** Paquetes Java con la implementación de la solución (incluye el paquete *test* asociado a las pruebas).

    - **Pruebas:**
        1. Diseño de los casos de prueba formato Cupi2 (escenarios y casos de prueba en formato Cupi2)

2. Este ejercicio puede realizarse individual o en parejas y debe tener en cuenta las siguientes condiciones:

    - Dentro del paquete de entrega debe haber un archivo y/o documento con los nombres de los integrantes. De lo contrario se le asignará la nota solo al estudiante que subió el ejercicio. En caso de ambos subirlo y no estar el documento, se le asignará la nota al que tenga la fecha más temprana pues se entiende que ese es el dueño original del ejercicio.

    - Ambos estudiantes son responsables del desarrollo total del ejercicio. Es decir, que cualquier separación interna de responsabilidades entre los miembros del grupo no justifica el desconocimiento del proyecto completamente. Si esto llegase a suceder entonces el ejercicio será invalidado para los dos estudiantes del grupo.

ENUNCIADO DEL PROBLEMA

Shrek y Fiona han tomado posesión recientemente del trono de Muy muy lejano y se encuentran en el proyecto de realizar un censo, para conocer el estado en que se encuentran sus habitantes. Ellos saben que en sus dominios pueden existir más de miles de personalidades, de manera que han considerado utilizar las herramientas tecnológicas para llevar a cabo el proceso.

Como es la primera vez que los funcionarios públicos cambiaran el papel por dispositivos electrónicos y software especializado, han contactado a su empresa de desarrollo para que les brinde apoyo en esta parte y se encargue por completo de la aplicación que se deberá utilizar para la importante tarea.

A continuación Shrek y Fiona describen lo que necesitan de la aplicación:

“Nosotros necesitamos primero una opción de registro, donde se pueda tomar la información de cada una de las personalidades del reino. Esperamos tener registrado nombre completo, fecha de nacimiento, número de identificación, sexo, tipo RH, dirección de residencia, teléfono, estado civil, cantidad de hijos, nivel educativo, profesión, estado laboral, salario (si el estado laboral es ‘desempleado’ el salario debe corresponder al valor devengado en su último trabajo). Y a medida que se vaya registrando una nueva personalidad, queremos que la información se mantenga organizada por edad, en alguna parte se agregue y se muestre una lista actualizada de todos los registros del momento.

Luego, como tendremos muchos funcionarios trabajando en el reino recolectando la información, necesitamos que de alguna manera se pueda ‘sacar los datos de registro’ de una aplicación y cargarla en otra (función de exportar e importar respectivamente), para poder hacer un análisis global de nuestros habitantes. Nosotros preferimos que esto se haga con la información en texto plano (la información de registro separada por punto y coma - ; -, organizados en el mismo orden en que fueron especificados), así también nos damos cuenta si la aplicación está ‘sacando’ efectivamente todos los registros que tiene.

Es muy importante que la aplicación controle que no se repitan registros, se deben tener en cuenta dos situaciones, cuando el registro lo hace el funcionario o cuando se realiza como una carga de información masiva de otro funcionario.

Cuando ya podamos tener todo cargado en una aplicación, queremos que se puedan realizar diferentes tipos de reportes, para empezar queremos: Listado de todos los hombres y listado de todas las mujeres organizados por tipo RH, listado de habitantes con hijos y estado laboral ‘desempleado’. Al generar los listados queremos que la aplicación nos dé la opción de guardarlo como un archivo reporte, para así consultarla cada vez que se necesite y poder compararla con los reportes de otras zonas.

Bueno, y para finalizar, Muy muy lejano es muy muy grande, la tarea de censar a nuestros habitantes les tomará a todos nuestros funcionarios cerca de 3 meses. Obviamente, no queremos que lo que registren en un día se borre cuando se les apague el equipo, esa información se tiene que mantener aunque se cierre la aplicación y continúen con el registro dos días después. Se espera que se utilice un mecanismo de almacenamiento eficiente y seguro, pues no queremos que personas no autorizadas puedan ver esta información.

Shrek es muy exigente, quiere todo ya y perfecto para que podamos volver a la vida del Pantano pronto, así que por favor esperamos lo mejor de ustedes en esta tarea."

Después de la entrevista con Fiona, Shrek añadió que para estar más seguros que la aplicación este correcta y completa, quisiera que se le añadiera al proyecto de desarrollo todo el análisis de pruebas unitarias.

RESTRICCIONES DEL PROBLEMA

1. Esta unidad evalúa el uso de pruebas unitarias para desarrollar programas correctos. Por lo tanto es indispensable esta implementación.

2. Esta unidad evalúa la definición e implementación de invariantes para la verificación de corrección de programas. Por lo tanto es indispensable su inclusión en la solución propuesta.

3. Esta unidad evalúa técnicas de ordenamiento, por lo tanto su solución debe implementar todas las técnicas (repartidas entre los diferentes requerimientos de ordenamiento) y debe ser explícita en la interfaz de usuario. Del mismo modo se evalúa la búsqueda binaria así que es obligatoria su implementación.

4. Esta unidad evalúa la implementación de persistencia en archivos de datos y archivos de texto en diferentes momentos de la solución. Se espera que usted lo desarrolle de acuerdo a lo visto en clase. Cualquier propuesta diferente deberá ser discutida previamente, acordada y aprobada por el profesor.

5. Esta unidad introduce elementos gráficos que deben ser incluidos dentro de su solución.

6. Esta unidad evalúa la implementación de excepciones como parte de su solución para el manejo y control de errores de ejecución.

RÚBRICA DEL EJERCICIO

Criterio	Porcentaje
Proceso de ingeniería de software	
Análisis *Estándar de la documentación* *Completitud de los requerimientos* *Asignación de Responsabilidades* *Cronograma de trabajo y roles*	*10%*
Diseño *Diagrama de clases del mundo del problema* *Diagrama de clases de la interfaz* *Diagrama de clases de las pruebas* *Diseño de pantallas y mapa de navegación*	*20%*
Implementación *Cumplimiento de los estándares de programación* *Cumplimiento de la documentación* *Cumplimiento de las restricciones del enunciado en cuanto a los temas de la unidad* *Funcionalidad en general*	*35%*

Pruebas *Estándar de la documentación* *Calidad de los escenarios y casos de prueba* *Funcionalidad en general de las pruebas*	*35%*
TOTAL	***100%***

Entrega 1:

Se deben considerar los siguientes ítems:

- **Planeación del proyecto:**
  1. Cronograma de trabajo y roles

     Significa la asignación de responsabilidades específicas a cada uno de los integrantes del grupo, detalle de las tareas a realizar y fechas de revisión entre el grupo de estudiantes. Estas últimas no incluyen la participación del docente y se sugiere considerar mínimo 1 por semana.
- **Análisis:**
  1. Documento de requerimientos funcionales y no funcionales con formato Cupi2.
- **Diseño:**
  1. Diagrama de clases del mundo y de la interfaz.
  2. Diseño de pantallas y Mapa de navegación

Entrega 2:

Se debe considerar añadir a la entrega 1 los siguientes ítems:

- **Diseño:**
  1. Diagrama de clases de las pruebas.

- **Implementación:** Paquetes Java con la implementación de la solución (incluye el paquete *test* asociado a las pruebas).

- **Pruebas:**
  1. Diseño de los casos de prueba formato Cupi2 (escenarios y casos de prueba en formato Cupi2)