

Objetivos del Curso

Este ejercicio busca trabajar los siguientes objetivos

Unidad 1	Unidad 2	Unidad 3	Unidad 4	Unidad 5
<p>Escribir el invariante de una clase e implementar los métodos necesarios para su verificación, utilizando la instrucción assert de Java.</p>	<p>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.</p>	<p>Escribir los algoritmos necesarios para manipular estructuras lineales que almacenan sus elementos enlazándolos entre ellos.</p>	<p>Construir interfaces de usuario que incluyan menús de opciones, gráficas animadas en 2 dimensiones y eventos del ratón como una mejora a la interacción del usuario con la aplicación.</p>	<p>Desarrollar un programa que maneje concurrencia, de manera que sea posible que ejecute más de una parte del programa de manera simultánea, utilizando hilos de ejecución (threads).</p>
<p>Desarrollar las clases y los métodos necesarios para implementar las pruebas unitarias automáticas, que ayuden a comprobar el correcto funcionamiento de un programa.</p>	<p>Manipular archivos de texto y utilizarlos para implementar algunos requerimientos del cliente.</p>	<p>Construir interfaces de usuario más complejas, utilizando nuevos distribuidores gráficos y nuevos componentes que van a facilitar el despliegue de información y la interacción con el usuario.</p>		<p>Aplicar los conceptos básicos de recursividad en la solución de problemas numéricos, búsqueda binaria y dibujo de fractales.</p>
<p>Implementar una interfaz de usuario que permita la interacción y despliegue de un grupo de objetos.</p>	<p>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.</p>	<p>Utilizar las interfaces Collection, List e Iterator de Java que permiten la manipulación abstracta de estructuras contenedoras.</p>		
<p>Reconocer la diferencia entre orden natural y orden parcial de los objetos por medio de la descripción de utilidades de las interfaces Comparable y Comparator.</p>	<p>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.</p>	<p>Utilizar la herencia como mecanismo de construcción de aplicaciones en Java y entender el papel que juega la clase Object en dicho lenguaje.</p>		
<p>Implementar algunos algoritmos clásicos de ordenamiento y búsqueda de información en estructuras de datos lineales y aplicarlos en la solución de un problema.</p>	<p>Utilizar estructuras enlazadas de objetos para modelar grupos de atributos no primitivos de longitud indeterminada.</p>	<p>Entender el uso que le hemos dado a la herencia en niveles anteriores, para construir interfaces de usuario y tipos de excepciones.</p>		
<p>Calcular el tiempo de ejecución de un algoritmo por medio de las operaciones de tiempo del sistema e implementar métodos que permitan generar muestras con datos aleatorios.</p>				

Enunciado del problema

Nombre del proyecto: El ataque de los bichos

Los horribles insectos están por todas partes! Pero tú cuentas con tus pantuflas para aplastarlos y hacerles la vida imposible. ¿Podrás hacerlo? No dejes uno sólo vivo. Adelante! Acábalos!

Una reconocida empresa de desarrollo de juegos, ha decidido sacar al mercado un juego para niños menores de 10 años llamado *El ataque de los bichos*. Este juego consiste en eliminar la mayor cantidad de bichos posibles que aparecen en pantalla durante un tiempo determinado dando click sobre ellos.

El juego cuenta con 5 niveles, cada uno más difícil que el anterior en donde los insectos irán apareciendo en pantalla para que el usuario pueda eliminarlos. Se tienen diferentes tipos de insectos de los que se desea almacenar una imagen, color predominante, puntaje, velocidad de avance, tiempo de metamorfosis y estado (huevo, larva o estado final) sin embargo cada insecto también cuenta con unas características especiales:

- Cucaracha: Numero de clicks necesarios para matarlas y si es voladora o no.
- Mosca: Tamaño y número de huevos puestos.
- Araña: Si es venenosa y calidad de su tela de araña (valor que indica fortaleza).
- Zancudo: Si transmiten o no enfermedades, sexo y de ser hembras la cantidad de personas que ha picado.

Adicionalmente, los creadores del juego desean guardar información del usuario para poder gestionar sus partidas futuras, así que desean conocer:

- Nombre del jugador, nivel máximo alcanzado, puntaje logrado.

La primera vez que el jugador inicia una partida se generan aleatoriamente 100 insectos, los cuales se van a desplegar en cada uno de los 5 niveles (20 insectos por nivel). De manera que cuando se vayan eliminando se actualice en pantalla el número de insectos eliminados y el puntaje acumulado. Cada nivel de la partida cuenta con un puntaje a alcanzar en un (1) minuto para poder avanzar permitiendo que la velocidad de avance de cada insecto incremente según su factor y el nivel en el que se encuentra. Se espera que en la etapa final de desarrollo, los insectos se desplieguen en pantalla y su desplazamiento se visualice de la parte baja de la ventana hasta la parte alta.

Una vez se ha terminado el juego, el programa debe mostrar las siguientes estadísticas:

- Cantidad de insectos generados discriminados por especie.
- Cantidad de insectos eliminados discriminados por especie.
- Puntaje del jugador por nivel: cantidad de eliminados x Puntaje insecto.
- Cantidad de clicks perdidos.

El juego llega a su fin en alguno de los siguientes casos:

- El jugador cierra la ventana (Se debe desplegar un mensaje de confirmación). Antes de cerrar se debe guardar la información de la partida en la que se encuentra.
- El jugador logra eliminar todos los insectos.
- Se termina el tiempo.

El desarrollo será incremental y se dividirá el proyecto en tres partes:

PARTE 1: Gestión de insectos

Se requiere que usted defina los intervalos válidos para cada uno de los atributos de los insectos. Luego de esto, se espera que genere aleatoriamente valores dentro de dichos intervalos para asignarlos a cada uno de los atributos de los 100 insectos que hacen parte de la partida. Además, todos los insectos generados se deben visualizar en una interfaz gráfica que permita identificarlos por el nivel al que pertenecen. Es importante aclarar que cada nivel debe contar por lo menos con tres clases de insectos.

Se requiere también que el juego permita ordenar a todos los insectos generados por velocidad de avance y por tiempo de metamorfosis. Finalmente, se le pide que la información de los insectos sea almacenada de forma permanente de manera que se pueda cargar cada que se ejecute esto es, el programa permite al usuario escoger el nivel que desea cargar. De tal manera que si el usuario ha finalizado el nivel 1 pueda empezar a jugar desde el nivel 2. El almacenamiento debe hacerse en un formato que solo sea reconocido por el programa que gestiona esa información. Finalmente, se quiere que toda la información de los insectos se pueda exportar en un archivo de texto plano. Debe existir la opción de ordenar los datos por velocidad de avance o tiempo de metamorfosis antes de escribir los datos en el archivo.

PARTE 2: Gestión de usuarios y lógica del juego.

El juego a desarrollar debe tener la capacidad de gestionar información (es decir, registrar, actualizar y eliminar registros) de usuarios. Cuando el usuario inicia el juego el programa debe estar en la capacidad de recordar información sobre su última partida o de registrar un nuevo jugador en caso de que el usuario no haya jugado antes. Para esta segunda parte, el programa debe ofrecer la funcionalidad de generar un reporte de los tres mejores puntajes almacenados. En este reporte se deben incluir: El nombre del jugador, su puntaje y la fecha en la que jugó por última vez. Debe tener en cuenta que para esta etapa puede ser necesario un rediseño del modelo del problema.

PARTE 3: Juego interactivo y entrega final

Esta última etapa contempla el cambio de la visualización de los insectos que se realizó durante la etapa 1 para pasar a la visualización del desplazamiento de estos en pantalla y su eliminación a través de clicks del mouse tal como se menciona en el enunciado y la correcta integración de las funcionalidades desarrolladas en la etapa 1 y 2 del proyecto.

RESTRICCIONES DEL PROBLEMA

PARTE 1

1. El desarrollo de su programa debe incluir pruebas unitarias automáticas de todos los métodos.
2. Todas las clases del mundo deben tener definidas y verificadas en sus respectivos métodos sus invariantes.
3. Debe implementar y utilizar al menos un (1) algoritmo de ordenamiento y al menos aplicar búsqueda binaria una vez.
4. Debe utilizar persistencia por serialización y por archivos de texto plano.
5. Debe incluir al menos tres elementos gráficos nuevos, es decir que no se utilizaron en AP1.

PARTE 2

Además de las anteriores el programa....

1. El programa debe hacer uso de estructuras lineales doblemente enlazadas para la gestión de usuarios
2. El programa debe realizar una búsqueda dentro de la estructura lineal doblemente enlazada.
3. El programa debe tener completa la lógica del juego, esto es, todos los requerimientos que permitan al usuario jugar (como manejo de tiempo , niveles, puntajes, entre otros) deben estar implementados.

PARTE 3

Además de las anteriores el programa....

1. El programa debe hacer uso de elementos gráficos usando el paintComponent.
2. El programa debe hacer uso de recursividad.
3. El programa debe hacer uso de hilos.

Publicación: xx de Febrero de 2015

Cada una de las tres partes tiene 2 entregas: Entrega documentación (parte A) y Entrega desarrollo completo de esa parte (parte B).

Listados de Entregas:

Entrega 1A: 10% - Viernes 27 de Febrero de 2015 – Documentación Parte 1

Entrega 1B: 25% - Viernes 13 de Marzo de 2015 – Desarrollo Completo Parte 1

Entrega 2A: 05% - Viernes 10 de Abril de 2015 – Documentación Parte 2

Entrega 2B: 25% - Viernes 24 de Abril de 2015 – Desarrollo Completo Parte 2

Entrega 3A: 05% - Lunes 4 de Mayo de 2015 – Documentación Parte 3

Entrega 3B: 30% - Viernes 15 de Mayo de 2015 – Desarrollo Completo Parte 3

Cada una de las entregas impares (documentación) debe tener lo que se lista a continuación:

Entregas Partes A

- **Planning - Planeación del proyecto:**

1. Cronograma de trabajo y los formatos PSP

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. Además de los formatos donde se registre su avance real en el desarrollo del proyecto.

Esta entrega es un enlace a un documento googledocs que su grupo actualizará durante el semestre con la información que recopile durante el proceso de desarrollo. Usted debe compartirlo con la opción de hacer comentarios o anotaciones, pero no con la opción de editarlo.

Debe compartirlo a:

lina.quintero@correo.icesi.edu.co si es del grupo 1

cevacu@gmail.com si es del grupo 3.

Importante: sólo debe compartir un enlace de cronograma y formatos PSP para todo el proyecto, lo que debe hacer en las entregas de las partes posteriores es completar la información de los documentos ya existentes.

2. Documento de requerimientos funcionales y no funcionales con formato Cupi2.

- **Design - Diseño:**

1. Diagrama de clases del mundo y de la interfaz gráfica.

- **Code - Código (Implementación):**

1. Implementación de la interfaz gráfica (sin lógica de la aplicación, solo las ventanas y los formularios).

Rubrica del ejercicio (Entrega Documentación)

Criterio	Puntos
Planning - Planeación del proyecto	Subtotal 20
<i>Cronograma de trabajo y Formatos PSP 5 puntos por cada formato.</i>	10
<i>Documento de requerimientos funcionales y no funcionales con formato Cupi2.</i>	10
Design – Diseño	Subtotal 55
<i>Estándar de la documentación</i>	5
<i>Compleitud de los requerimientos</i>	10
<i>Diagrama de clases del mundo del problema</i>	25
<i>Diagrama de clases de la interfaz</i>	15
Code - Código (Implementación)	Subtotal 25
<i>Implementación de la interfaz gráfica</i>	25

TOTAL	100
--------------	------------

Entregas Parte B

Usted debe entregar: los mismos ítems de la entrega documentación (si les ha realizado modificaciones o no) **más** los siguientes:

- **Code - Código (Implementación):**
 1. Paquetes Java con la implementación de la solución (incluye el paquete test asociado a las pruebas).
- **Design – Diseño:**
 1. Diagrama de clases del mundo, de la interfaz y de las pruebas.
- **Test – Pruebas:**
 1. Diseño de los casos de prueba formato Cupi2 (escenarios y casos de prueba en formato Cupi2)

Rubrica del ejercicio (entrega desarrollo completo)

Recuerde que en esta segunda entrega debe enviar su proyecto completo, con todas las partes detalladas a continuación. Si usted no modificó ninguna de las partes enviadas en la entrega documentación, de igual modo debe enviarlas con esta entrega y serán calificadas nuevamente aunque esta nueva ponderación solamente será tomada en cuenta en la nota de la segunda entrega. La nota de la Entrega Documentación no cambia. Por tanto, si usted actualizó alguna de estas partes y desea que le sean tomadas en cuenta estas mejoras, no olvide incluir los documentos actualizados (no los anteriores).

Criterio	Puntos
Planning - Planeación del proyecto:	Subtotal 10
<i>Cronograma de trabajo y Formatos PSP</i>	5
<i>Documento de requerimientos funcionales y no funcionales con formato Cupi2.</i>	5
Design – Diseño	Subtotal 21
<i>Estándar de la documentación</i>	2
<i>Complejidad de los requerimientos</i>	2
<i>Diagrama de clases del mundo del problema</i>	6
<i>Diagrama de clases de la interfaz</i>	3
<i>Casos de uso y Diagrama de clases de las pruebas</i>	8
Code - Código (Implementación)	Subtotal 59
<i>Cumplimiento de los estándares de programación</i>	4
<i>Cumplimiento de la documentación</i>	4
<i>Cumplimiento de las restricciones del enunciado en cuanto a los temas de la unidad</i>	8
<i>Funcionalidad en general de la aplicación</i>	25
<i>Funcionalidad en general de las pruebas</i>	15
Test – Pruebas:	Subtotal 15
<i>Estándar de la documentación</i>	5
<i>Calidad de los escenarios y casos de prueba</i>	8
TOTAL	100

Nota: El proyecto puede hacerse de forma individual o en parejas.