

Programación con Objetos (C1) - Trabajo Final

1. Introducción

Este documento presenta los detalles del Trabajo Final, que junto con la aprobación del examen parcial es uno de los requisitos para la aprobación de la materia.

El trabajo tiene las siguientes características:

- Consiste en el diseño y la implementación de un modelo para un problema determinado.
- Deberá desarrollarse en grupos de 2 personas.
- Deberá desarrollarse utilizando las horas de práctica de manera virtual, además de cierto tiempo extra fuera de clase.
- Deberá desarrollarse de manera iterativa e incremental utilizando la técnica de Test-Driven Development (TDD).
- El enunciado será designado por el equipo docente.
- Los docentes realizarán el seguimiento de cada trabajo, respondiendo las consultas que surjan, y podrán solicitar entregas parciales para validar el trabajo realizado hasta el momento.
- El resultado del trabajo final es la implementación, tests y el código resultante, y la elaboración de un informe que incluya el modelo y decisiones de diseño.

2. Criterios de Evaluación

Cada grupo debe entregar su proyecto completo. La calificación final del trabajo será **individual** y dependerá de los siguientes criterios de evaluación:

- El diseño y la implementación. El modelo definido, la aplicación de los conceptos y metodologías aprendidas durante el curso, así como que el trabajo cumpla o no con las especificaciones mínimas requeridas.
- El tiempo de trabajo en clase donde se utilizará no sólo para realizar consultas sino para discutir acerca del diseño, y programar de a pares para ir resolviendo el trabajo.
- La calidad de la documentación. El trabajo entregado deberá cumplir con ciertos requisitos mínimos respecto a su documentación (detallados más abajo).

Tener en cuenta que todos los criterios de evaluación son importantes para definir la nota final del mismo.

2.1. Formato del Informe

El informe del trabajo desarrollado debe contener como mínimo las secciones descritas abajo. Además puede incluir cualquier información que considere relevante.

- Carátula: con enunciado asignado, integrantes del grupo, materia y fecha.
- Alcance: Escriba de 5 a 15 renglones
 - Qué partes del enunciado se implementaron, cuáles no y por qué.
 - En el caso de agregar alguna característica extra a las que figuraban en el enunciado, enumérelas y explique por qué se agregaron.
- Modelo: Describa cuáles son los objetos más importantes de su sistema, y sus principales responsabilidades.
- Progreso: Explique cómo se fue construyendo iterativa e incrementalmente el TP a lo largo de su realización.
- Dificultades: Describa qué dificultades tuvo para implementar el TP. Especifique qué partes fueron más difíciles y cuáles fáciles. Puede incluir las dificultades del testing.
- Conclusiones: Escriba las conclusiones que haya obtenido a partir de este trabajo. Puede describir por ejemplo qué conceptos de los aprendidos utilizó en mayor medida y cuáles considera que no pudo aplicar.

2.2. Entrega

La fecha de entrega se comunicará por mail a la lista de estudiantes. La entrega del Trabajo Final completo se subirá al Campus con el asunto. Todo el contenido de la entrega debe ir en un archivo comprimido en formato .zip, y con nombre de acuerdo a la convención TrabajoFinal-Entrega-GrupoX donde X es el número de grupo asignado.

3. Enunciados

Los enunciados sirven a modo de guía para comprender el dominio y saber lo que se espera de ellos, pero de ser necesario los docentes podrán ajustar el alcance, si eso ayuda a una mejor evaluación.

Cada grupo tendrá asignado de manera aleatoria uno de los siguientes cuatro enunciados:

TEQ

Plan Táctico y Estratégico de Quilmes es un juego de mesa de 2 a 4 jugadores. Plantea un conflicto bélico que ocurre sobre las nueve localidades que componen el partido de Quilmes. Al comienzo de la partida los jugadores se reparten todas las localidades, que son ocupadas con fichas del color asignado a cada jugador.

Todos los participantes tienen un objetivo general, consistente en ocupar 6 localidades. El jugador que primero logre cumplir este objetivo será el ganador del juego.

Las reglas más importantes son las siguientes:

- El primer jugador en adueñarse de 6 localidades gana.
- Cada localidad debe pertenecer a un solo jugador. Es decir, solo pueden haber tropas (como mínimo una) de su color en dicha localidad.
- Los jugadores pueden adueñarse de las localidades de otros jugadores atacando a las tropas estacionadas en ellas. Para poder atacar deben cumplirse dos cosas: la localidad atacante debe tener dos o más tropas, y una localidad solo puede atacar a localidades limítrofes.
- Cada vez que un jugador gana una localidad nueva puede tomar una tarjeta del mazo de localidades. Si la localidad en esa tarjeta le pertenece, la tarjeta se remueve del mazo y el jugador agrega una tropa adicional a la localidad en cuestión. Si la localidad NO le pertenece, la tarjeta se vuelve a poner en el mazo y este se mezcla.
- Una vez todos los jugadores hayan tenido su turno de atacar pueden agregar más ejércitos, dependiendo de la cantidad de localidades que posean. Un jugador con una sola localidad tiene permitido agregar 2 ejércitos. Un jugador con dos a cuatro localidades puede agregar tantos ejércitos como localidades tenga. Un jugador con cinco localidades puede agregar 4 ejércitos solamente.
- Debe ser posible imprimirse en un archivo el estado de todos los jugadores, con sus colores correspondientes, las localidades que posee cada uno y los ejércitos estacionados en cada localidad.

Los ataques entre localidades ocurrirán mediante tiradas de dados. En cada situación de batalla hay una tirada atacante y una tirada defensora, cada cual puede tener entre 1 y 3 dados dependiendo de sus tropas.

Las reglas de las tiradas son las siguientes:

- Dos tropas conceden a una localidad un dado para atacar. Tres tropas conceden dos dados. Cuatro o más tropas conceden siempre tres dados.
- Una tropa concede a una localidad un dado para defender. Dos tropas conceden dos dados. Tres o más tropas conceden siempre tres dados.
- Cuando ocurre un ataque, ambas tiradas se ordenan de mayor a menor y se comparan para ver cuantos ejércitos perdió cada uno.

Por ejemplo, supongamos que el atacante tira con 3 dados y el defensor también. Dada la siguiente tirada:

Atacante: 6 4 1

Defensor: 6 3 2

El atacante pierde 2 ejércitos (los correspondientes al 6 y 1). El defensor pierde un ejército (el correspondiente al 3). Cuando los dados son iguales el que pierde es el atacante.

Dada la siguiente tirada:

Atacante: 5 3

Defensor: 4

El atacante no pierde ningún ejército y el defensor pierde uno.

El resultado de la tirada debe imprimirse en un archivo, de forma que pueda leerse qué dados tiró cada jugador y el resultado final.

Carmen Sandiego

Carmen Sandiego es un juego donde un objeto de valor histórico es robado de un país y el detective debe encontrar al delincuente, persiguiéndolo en diferentes países del mundo. El juego comienza en el país donde el delito fue cometido. Para ello el detective puede consultar una lista de sospechosos (se sabe que uno de ellos cometió el delito), y los países a dónde puede viajar (se puede asumir que hay un conjunto fijo de países, conocidos al inicio del juego). Además el detective cuenta con un límite de tiempo para encontrar el delincuente (que podemos medir en cantidad de horas).

Un detective podrá obtener una o dos pistas de cada testigo en cada país que esté (comenzando por donde fue cometido el delito). Los testigos son tres, una de las pistas le indicará a qué país viajar para perseguir al delincuente y las otras dos le otorgará datos del mismo, si el detective equivoca el destino y viaja a un país donde el ladrón nunca estuvo, los testigos sólo dirán que no vieron a ningún sospechoso. Cada vez que el detective interroge a un testigo, perderá dos horas. Cuando viaje a otro país, perderá 10 horas. Cuando el detective obtiene una pista sobre el aspecto del delincuente, debe recordarla en algún lugar ya que las mismas pueden servir para filtrar posibles sospechosos; si con los datos introducidos hasta el

momento hay un sospechoso que concuerde con la descripción, podrá emitir la orden de arresto para ese sospechoso. De los sospechosos están los siguientes datos:

- Color de ojos
- Forma del cabello (pelado, corto, largo, lacio, ondulado)
- Color de cabello
- Color de tez
- Contextura física (delgado, normal, relleno)
- Sexo
- Altura (baja, media, alta)

Implemente el Carmen Sandiego teniendo en cuenta que:

- Se debe poder obtener una lista de sospechosos dadas unas características (obtenidas de las pistas).
- Puede que no se requieran todas las pistas para arrestar a un sospechoso. Por ejemplo, si la primera pista que obtiene el detective es que el culpable tiene el cabello negro y solo un sospechoso tiene cabello negro, la orden de arresto se puede emitir inmediatamente.
- Debe decrementarse el tiempo restante cuando se viaja o se interroga según corresponda.
- Si el detective se queda sin tiempo, la partida se considera perdida.
- Si el detective obtiene la orden de arresto para el delincuente, la partida se considera ganada.
- Cada testigo que se interroga debe poder otorgar una pista no repetida.
- Cada país tiene un objeto de valor que puede ser robado.
- Existen distintos detectives: El clásico, el persuasivo y el de Interpol.
El clásico obtiene 1 pista al interrogar un testigo y pierde 2 horas, cuando viaja a otro país pierde 10 horas.
El persuasivo es hábil interrogando testigos, a cada testigo le saca 2 pistas y sólo pierde 1 hora interrogando.
El de Interpol tiene más contactos y pasaportes por lo que sus viajes a otros países se realizan en menor tiempo. Tarda sólo 8 horas.
- Existen países limítrofes. Si el detective viaja a un país limítrofe, su tiempo de viaje se reduce en 3 horas o sea que tarda 7.
- Tiene que ser posible imprimir la orden de arresto en un archivo. Esto incluye el nombre del culpable y todas sus características.

Biblioteca

La biblioteca de la Universidad necesita renovar su aplicación para manejar los socios, el inventario y los préstamos que se otorgan. De los socios se necesita saber su nombre y apellido, su número de legajo y su fecha de registración. Cada socio debe renovar su membresía anualmente, de lo contrario no podrá pedir prestados libros. Cada libro posee su número de inventario, el autor, título y una lista de temas relacionados.

Las reglas que rigen los préstamos son las siguientes:

- No se pueden prestar libros a un socio cuya registración/renovación expiró.

- Un socio no puede pedir prestados más de 5 libros al mismo tiempo. Debe devolver alguno antes de pedir prestado otro.
- En general, los libros se prestan durante una semana, pero existen ejemplares que sólo pueden ser pedidos y devueltos el mismo día.
- Los préstamos pueden renovarse por una semana, siempre y cuando la renovación se realice durante la semana del préstamo.
- Existen los préstamos extendidos (determinados por una fecha de inicio y otra de fin), cuya duración es hasta la fecha de fin inclusive. (No se pueden renovar)
- La biblioteca puede verificar si se debe sancionar un socio. Debe sancionar un socio si no devolvió un libro en el plazo estipulado. Se debe repetir este mismo proceso para todos los socios inscriptos.

Implemente además los siguientes listados que se le solicitan al sistema:

- Libros más pedidos: retorna una colección ordenada comenzando por aquellos libros que han sido prestados durante más tiempo total. Si existen varios ejemplares del mismo libro, deben sumarse los días todos los ejemplares. El listado NO debe contener libros que nunca hayan sido prestados.
- Socios solidarios: un socio solidario es aquel que devuelve los libros rápidamente, para que otros puedan sacarlo. El listado debe incluir sólo a aquellos socios que cuyo tiempo promedio de devolución de libros sea menor a 3 días.

Proponga 2 listados más e implementelos. Genere un archivo con el contenido de alguno de los listados.

Mini-Terminal Unix

Se desean simular determinadas características de un sistema operativo Unix. Para ello, se debe modelar una estructura de archivos *a-la* Unix, junto con ciertos comandos del sistema operativo que permiten manipular a los archivos.

El sistema de archivos debe permitir tener archivos y directorios. Existe un directorio destacado que es el raíz. Cada directorio puede contener archivos, u otros directorios. El usuario actual puede estar posicionado en un directorio a la vez. Inicialmente es el raíz, pero puede ir a un subdirectorio usando el comando `cd`. Para volver al directorio padre, debe usar el comando `cd ..`. Para ver el contenido del directorio actual, puede usar el comando `ls` (que va a retornar todos los archivos y subdirectorios que haya).

Además existen usuarios, que tienen un nombre y una contraseña. Existe un usuario administrador conocido como `root`.

El sistema debe contemplar:

- Iniciar sesión como `root` o como cualquier usuario. Se debe lanzar un error si el usuario no existe, o la contraseña es incorrecta.
- Cada vez que se inicia sesión, el usuario debe comenzar en el directorio raíz.
- Agregar o remover usuarios (sólo `root` puede hacerlo)

- Crear y borrar archivos y directorios en el directorio actual.
- El archivo tiene un contenido textual que puede ser editado en cualquier momento. Nos interesa saber si un usuario puede editar un archivo o no.
- Navegar entre los diferentes directorios (tal como lo hace el comando `cd`)
- Listar el contenido de un directorio (tal como lo hace el comando `ls`)
- Administrar los permisos de archivos y directorios:
 - Tenemos permisos de lectura, escritura y ejecución (para los archivos) y de lectura y escritura (para los directorios)
 - A cada archivo se le asignan permisos para el usuario *owner* (o sea, el usuario logueado que creó el archivo), y para los demás.
 - Los permisos deben ser chequeados al momento de una operación.
 - El usuario *root* puede ejecutar cualquier operación con los archivos.
 - El *owner* de un archivo o directorio siempre tiene permisos de lectura y escritura
 - Ejemplo: si el archivo `tesis.txt` es propiedad de Juan, y tiene permiso sólo de lectura para los demás, entonces Juan (y *root*) pueden borrar el archivo (porque ambos tienen permiso de escritura), pero el usuario Pedro sólo puede leer su contenido (porque es de los “demás”)
- Nos interesa copiar un archivo a la carpeta de Cuis, utilizando streams.
- Se pide además que se tenga la posibilidad de enviar el resultado del comando `ls` a un archivo en el disco real (como lo hace el operador `>`)
 - La salida esperada tiene la siguiente forma:

miDirectorio:

miSubDirectorio	lectura	juan
miArchivo.txt	lectura/escritura	root
miArchivo2.txt	lectura/escritura/ejecucion	pepe