PRÁCTICA 3

Tecnología de la Programación

Ingeniería Informática 2008/2009

Resumen

En esta práctica continuaremos con el desarrollo del simulador del sistema prepago de la tarjeta Millenium. Se trabajará en el desarrollo de un banco de pruebas que nos ayude a validar el correcto funcionamiento de nuestro programa y detectar los errores de funcionamiento, así como documentar correctamente las clases y métodos.

Objetivos

- Documentar de forma correcta el código fuente.
- Definir un conjunto de datos y baterías de pruebas.
- Comprender el concepto de cobertura de código.
- Utilizar **ant** para la compilación del programa y la ejecución de las pruebas (OPCIONAL)

Material

Para la realización de esta práctica partiremos del código desarrollado durante la práctica 2, se adjuntan nuevas versiones de algunas de las clases para permitir la ejecución del interprete de forma no interactiva. Las nuevas clases se encuentra subidas en la herramienta de teleformación utilizada por la asignatura.

Documentación adicional

Página principal de Javadoc

http://java.sun.com/j2se/javadoc/

How to Write Doc Comments for the Javadoc Tool
http://java.sun.com/j2se/javadoc/writingdoccomments/index.html

Página principal de Ant

http://ant.apache.org/

Manual de Ant

http://ant.apache.org/manual/index.html

Descripción del trabajo a realizar

Para la realización de esta práctica se partirá del resultado final de la práctica anterior P2, para ello se copiará (svn copy) el directorio P2 de la práctica anterior en un directorio P3.

La primera parte de la práctica consistirá en documentar siguiendo las normas el código fuente utilizando JavaDoc.

Se sustituirán los ficheros con las nuevas clases proporcionadas, para la interfaz de usuario. En esta práctica modificaremos el código de **FactoriaSistemaPrepago**. En esta clase se proporciona el método *public static ISistemaPrepago crearSistemaPrepago(String nombre)* que permite la creación de distintos SistemaPrepago previamente inicializados.

Para la definición de los bancos de prueba iniciales, esta clase se debe modificar para poder utilizar el SistemaPrepago con datos iniciales ya precargados (Tarjetas,Estaciones, Operaciones).

La utilización de estos SistemasPrepago será posible utilizando un parámetro con el nombre del mismo que se quiera cargar (TarjetaA,TarjetaB,TarjetaC, ….). Están definidos inicialmente 3 pero se pueden definir los que el alumno estime necesario.

Es misión del alumno definir tres conjuntos de datos iniciales para la batería de pruebas de validación que se realizarán como parte de esta práctica. Sirva de ejemplo el siguiente código:

```
if (nombre.equals("TarjetaA")) {
        sistema.crearTarjeta(“1234567”,3000,”Juan Gomez”,”social”);
        sistema.crearEstacion(“1”,”CENTRO”,”PAJARITAS”);
        // ESTO ES UN EJEMPLO DE DATOS BASE PARA PRUEBAS
        return sistema;
}
```

La actual versión del **Main** proporcionada permite una ejecución del entorno no interactiva, de forma que será posible la utilización de ficheros para realizar conjuntos de operaciones definidas en los mismos. Para ello habrá que pasarle el siguiente parámetro “-n” al llamar a Main.

El siguiente ejemplo ejecutaría de forma no interactiva el conjunto de datos de TarjetaA
$java -cp class/ iu.Main -n TarjetaA

El alumno deberá diseñar y definir baterías de pruebas que realicen una cobertura del 100% de las líneas de código desarrolladas por el alumno. Una batería de pruebas queda definida en el entorno de esta práctica como el conjunto formado por los datos iniciales, una serie de ordenes almacenadas en un fichero de texto que se cargarán por la entrada estándar y el resultado esperado de la ejecución del código. Y el comando que ejecuta la prueba (**OJO ajustar las rutas a vuestra estructura de directorios**):

$java -cp class/ iu.Main -n tarjetaA < Bateria1Ordenes.txt > Bateria1Salida.txt

Para obtener la cobertura del 100% el alumno sólo podrá ejecutar baterías contra el interprete de comandos, se podrán detectar métodos que no son nunca utilizados, o ramas de condiciones que no son nunca ejecutadas, o funcionamientos incorrectos o no contemplados de la aplicación.

Para que los resultados puedan ser comparables y evaluables es necesario que sea homogénea la presentación por pantalla de los resultados, utilizando una función toString común para todos:

---

*DATOS TARJETA:
Id: $ID_TARJETA
Nombre: $NOMBRE_TARJETA
Saldo: $SALDO_TARJETA
Tipo: GENERAL/SOCIAL/UNIVERSITARIA

---

*DATOS ESTACION:
Id: $ID_ESTACION
Zona: $ZONA_ESTACION
Nombre: $NOMBRE_ESTACION

---

*DATOS OPERACION:
Tarjeta: $ID_TARJETA
Fecha: dd/mm/aaaa
Estacion: $ID_ESTACION
Tipo: FICHAR ENTRAR|FICHAR SALIR
Importe: $IMPORTE_OPERACION

---

*DATOS OPERACION:
Tarjeta: $ID_TARJETA
Fecha: dd/mm/aaaa
Tipo: CARGA
Importe: $IMPORTE_OPERACION

---

*DATOS OPERACION
Fecha: dd/mm/aaaa
Tarjeta Origen: $ID_TARJETA_ORIGEN
Tarjeta Destino: $ID_TARJETA_DESTINO
Tipo: TRANSFERENCIA
Importe: $IMPORTE_OPERACION

La notación obliga que después de cada dos puntos siempre hay un espacio, y no se acentúan las palabras.

En la memoria debéis describir los resultados obtenidos con el plan de pruebas, se recomienda ir documentando a medida que vais tomando decisiones, se considera un éxito que hayáis encontrado errores en el funcionamiento de vuestra práctica. Debéis documentar cada vez que encontréis un error, cada error lo debéis documentar de la siguiente forma y numerar de forma correlativa.

ERROR NUMERO XX:
Batería de pruebas: bateriaXXX
Fecha detección: dd/mm/aaaa
Descripción error: XXXXXXXXXXXXXXXXXXXXXXX
Descripción solución:
Validada batería de pruebas: (SI/NO)

Los errores detectados (BUGS) los iréis anotando en un fichero errores.txt en el directorio P3/tests/errores.txt, que subiréis al Subversión cada vez que realicéis un cambio o detectéis un error.

Cuando detectéis un error guardar los resultados de esa prueba, corregir el programa y repetir la prueba con los mismos datos y comprobar que la se ha solucionado el problema.

Entregables:

- El código documentado de todas las clases del paquete model
- Las distintas baterías de pruebas (fichero de entrada, fichero de ejecución, fichero resultado ejecución y fichero describiendo los objetivos de la batería de pruebas.
- El fichero build.xml para la parte opcional de ANT.
- Memoria describiendo el trabajo realizado detallando como se han inicializado los diferentes Bancos de pruebas, así como el proceso de detección de errores y su corrección.
- Documentación javadoc generada en la carpeta doc/javadoc/

La entrega de las prácticas se realizará a través del repositorio Subversion de cada grupo de prácticas, con la siguiente estructura:

/trunk/P3
    src/ (código fuente)
    test/bateria1 (fichero entrada, fichero resultante, descripción de pruebas realizadas y que partes del código cubre la batería de pruebas)
    test/bateria2
    .
    test/bateriaN
    doc/tp-gXXX-p3-memoria.pdf (donde XXX es el número de grupo)
    doc/errores.txt (documentación detección de BUGs)
    doc/javadoc/ (documentación javadoc generada)

Para la organización de los ficheros de batería de pruebas se podrá utilizar como ejemplo los siguientes nombres de ficheros:

Bateria1Descripcion.txt
Bateria1Ordenes.txt
Bateria1Resultados.txt
Bateria1.sh

Donde en Bateria1.sh se encuentra el siguiente código de scripting para la ejecución de la batería de pruebas

```
#!/bin/sh
java -cp ../../class/ iu.Main -n TarjetaA < Bateria1Ordenes.txt > Bateria1Resultados.txt
```

Es recomendable incorporar un script en el directorio test que ejecute todos los scripts de las baterías de pruebas.

Plazo de entrega

La duración estimada para la realización de la práctica es de dos semanas lectivas, por tanto la fecha de entrega es el **01/05/2009** pero el enunciado de la **práctica 4 estará disponible desde el 27 de abril de 2009**.