ELO330: Programación de Sistemas: Segundo Certamen Parcial 6/11/2002

1. Para el programa listado:
a) ¿ Explique qué hace? y ¿Cuál sería su salida si se ejecuta con: java NewIODemo prueba.txt ?
Si prueba.txt tiene contenido:
"Ojalá a todos le vaya bien,
que sus trabajos y proyectos tengan éxito
y pasen un buen fin de año."

El programa lee línea por línea el archivo de entrada y lo escribe en el archivo de salida IODemo.out. En el archivo IODemo.out, cada línea es iniciada con un número correlativo de línea.

La salida seria:

Line 1 Ojalá a todos le vaya bien,
Line 2 que sus trabajos y proyectos tengan éxito
Line 3 y pasen un buen fin de año.

b) Hay algún cambio en la salida del programa si reemplazamos la línea en negrita por:
while((s = in.readLine()) != null)
Justifique su respuesta.

Si. En este caso la salida mostraria:
Line 3Ojalá a todos le vaya bien,
Line 3que sus trabajos y proyectos tengan éxito
Line 3y pasen un buen fin de año.

Esto se debe a que in es un objeto que maneja un buffer, por lo tanto la lectura de la primera línea gatillara la lectura de las tres líneas por larte de li, luego su valor se mantendrá en 3 mientras se obtengan las otras dos líneas desde in.

```
//---------------------------------------------//
import java.io.*;
public class NewIODemo {
  public static void main(String[] args) {
    String s;
    try {
        LineNumberReader li = new LineNumberReader(new FileReader(args[0]));
        BufferedReader in =  new BufferedReader(li);
        PrintWriter out = new PrintWriter(new BufferedWriter(new FileWriter("IODemo.out")));
       while((s = li.readLine()) != null )
              out.println("Line " + li.getLineNumber() + s);
       out.close();
    } catch(IOException e) {
       System.out.println("IO Exception");
    }
  }
}
```

---

2. a) ¿Qué es la programación conducida por eventos (Event-driven-programming)?

La programación conducida por eventos es aquella donde en programa esta preparado para reaccionar a múltiples vías de entradas en forma concurrente. A diferencia de la programación secuencial tradicional donde típicamente hay solo una fuente de datos de entrada, en esta programación se deben definir segmentos de programas asociados a cada una de las múltiples posibilidades de la entrada. Como analogía

se puede decir que este modelo de programación es una abstracción de alto nivel para las interrupciones de hardware.

b) Dé un ejemplo de sistema cuya programación sea conducida por eventos.
Una interfaz gráfica de usuario en la cual al múltiples botones que pueden ser presionados. Cada botón gatilla una acciones diferente.

c) Explique qué elementos deben ser configurados para atender a un evento. Presente un ejemplo específico. (no se requiere codificar con detalles sintácticos)

Debe haber un objeto que sea capaz de recibir eventos, este objeto debe tener registrado otro objeto que se haga cargo de responder al evento recibido.
Para lograr desacoplar el diseño, normalmente los objetos que se hacen cargo de las acciones a ser desarrolladas son instancias de una clase que implementa la interfaz correspondiente a el o los eventos o deriva una clase que tiene definida funcionalidad vacía para cada evento permitiendo que la clase derivada sobremonte los métodos de interés.

Para que lo anterior funciones se debe tener entonces:
i) Crear un objeto que recibirá el evento.
ii) Registrar en él el objeto que debe ser llamado ante la llegada del evento.
iii) Definir la clase que dará originen a objetos capaces de atender los eventos.

Ejemplo especifico:

a) Defino una clase que implemente la interfaz del evento que me interese atender. Esta es la clase "escuchadora" (listener).
b) Declaro un botón,
c) Registro en el botón un instancia de mi clase escuchadora.
d) El botón es incorporado en un elemento gráfico.

d) Cuando un evento tiene lugar, ¿cuál es el orden de las acciones o flujo del programa que tiene lugar cuando se atiende el evento? Desarrolle la parte dinámica cuando a su ejemplo de c) le llega un evento.

a) Un usuario presiona el botón
b) La máquina Java detecta esta acción y busca si hay objetos gráficos que deban ser notificado.
c) La máquina detecta que e botón debe ser notificado del evento y accede al objeto que se ha registrado para atender el evento.
d) Java invoca el método del objeto registrado.
e) Se ejecuta el método y el control vuelve a la maquina virtual para atención de otros eventos.

No es preciso que todos estos pasos detallados sean listados para obtener una respuesta buena.

---

3. Se tienen las siguientes clases:

```
import java.util.*;
class Manzana {
 Manzana() {}
  void print() {  System.out.println("Yo soy una manzana");  }
}
class Naranja {
   Naranja() {}
  void print() {   System.out.println("Yo son una Naranja");  }
}
public class Frutas {
  public static void main(String[] args) {
    ArrayList frutas = new ArrayList();
    for(int i = 0; i < 7; i++)
      frutas.add(new Manzana());
    frutas.add(new Naranja());
```

```
    for(int i = 0; i < frutas.size(); i++)
      frutas.get(i).print();
  }
}
```

a) Indique si el programa posee errores de compilación. ¿Cuáles serían y cómo los corregiría haciendo cambios mínimos si los hay?

El programa presenta un error de compilación debido a la invocación del método print() sobre en objeto retornado por get(i) al final. Debe haber un cambio de clase antes de proceder a invocar el método. El error de compilación se puede corregir reemplazando la última sentencia por:
((Manzana) frutas.get(i)).print();

b) ¿Cuál es o son los errores de ejecución -luego de sus cambios en a- y cómo los corregiría si los hay?

Hay un error de ejecución al intentar acceder al método print de una manzana cuando en realidad el objeto almacena una Naranja.
Se puede corregir definiendo una clase superior, por ejemplo Fruta. El programa queda:

```
import java.util.*;

class  Fruta {
        void print() {};
}
class Manzana extends Fruta{
 Manzana() {}
  void print() {  System.out.println("Yo soy una manzana");  }
}
class Naranja extends Fruta{
   Naranja() {}
  void print() {   System.out.println("Yo son una Naranja");  }
}
public class Frutas {
  public static void main(String[] args) {
    ArrayList frutas = new ArrayList();
    for(int i = 0; i < 7; i++)
      frutas.add(new Manzana());
    frutas.add(new Naranja());
    for(int i = 0; i < frutas.size(); i++)
      ((Fruta) frutas.get(i)).print();
  }
}
```

---

4. a) Desarrolle un applet que escriba en pantalla: "Yo no hago nada"
El mensaje debe ser persistente, es decir reaparecer luego de minimizar y restaurar la ventana del navegador.

```
import java.lang.*;

class pruebaApplet extend Applet {
        void print(Graphics g) {
                g.drawString("Yo no hago nada", 20,20);
        }
}
```

b) Muestre el archivo html que permitiría desplegar su applet en una página WEB.

```
<applet class=pruebaApplet.class width=500 height= 200>
</applet>
```