

Problemas para el Control 2 – CC1001-03, año 2008

Control 1 CC10A 2001 – Problema 2

a) Escribir un método (función) de encabezamiento

```
static public String borrar(String x, String y)
```

que elimine todas las apariciones de x en y. Por ejemplo, la invocación `borrar("bra","abracadabra")` entrega el string "acada".

b) Utilice el método anterior en un programa que lea todas las líneas de un archivo y las grabe en otro archivo eliminando todas las apariciones de los strings ":-)" y ":- (" . El programa debe obtener el nombre del archivo de entrada siguiendo el diálogo indicado en el siguiente ejemplo:

```
Nombre de archivo de entrada ? carta
```

El resultado quedará en el archivo carta.out

Control 1 CC10A 2001 – Problema 3

Para operar con un número secreto generado por el computador se dispone de la clase `Secreto` con los métodos indicados en la siguiente tabla:

| encabezamiento | significado | ejemplo |
|-----------------------------------|--|---|
| <code>Secreto(int x,int y)</code> | Constructor que calcula y guarda un N° secreto (entero entre x e y generado al azar) | <code>Secreto s = new Secreto(10,20)</code> |
| <code>int comparar(int x)</code> | -1 si N° secreto < x, 0 si N° secreto = x, 1 si N° secreto > x | <code>s.comparar(15)</code> |
| <code>int menorMasCerca()</code> | Entrega el N° menor más cercano que ha sido comparado con el N° secreto | <code>s.menorMasCerca()</code> |
| <code>int mayorMasCerca()</code> | Entrega el N° mayor más cercano que ha sido comparado con el N° secreto | <code>s.mayorMasCerca()</code> |

a) Escribir un programa que utilice la clase anterior para implementar un juego en que el usuario tenga que adivinar un N° secreto generado por el computador, siguiendo las reglas y el diálogo indicado en el siguiente ejemplo:

```
adivine un N° X entre 1 y 100
```

```
X ? 27          pregunta
```

```
X < 27          respuesta
```

X ? 15
X > 15

X ? 0 solicitar ayuda
15 < x < 27 responder con menor y mayor más cercanos
..

X ? 25
X = 25 Felicitaciones

b) Escriba las instrucciones del método comparar, considerando la siguiente declaración de la clase Secreto:

```
class Secreto {
    private int secreto, menor, mayor;
    public Secreto(int x,int y) { //constructor
        secreto = x + (int) (Math.random() *(y-x+1));
        menor = x - 1;
        mayor = y + 1;
    }
    public int menorMasCerca(){ return menor; }
    public int mayorMasCerca(){ return mayor; }
    public int comparar(int x){ . . .}
}
```

Control 1 CC10A 2002 – Problema 2

A) Escribir la función (método) **static public String eliminar(String X,String Y,String Z)** que elimine del string **X** todos los substrings comprendidos entre los delimitadores **Y** y **Z**. Por ejemplo, **eliminar("a]b[cd]e[f]g[h","[","]")** entrega **"a]beg[h"**. En caso que el segundo delimitador sea un string vacío, se deben eliminar todos los caracteres a partir del primer delimitador. Por ejemplo, **eliminar("ab--cd","--","")** entrega **"ab"**.

B) (Ponderación: 1/3) Utilice la función **eliminar** en un programa que lea líneas y las escriba eliminando los substrings de la forma **<caracteres>** y **%caracteres**. Por ejemplo, **"ana <maria>rios%gonzalez"** debe escribirse **"ana rios"**. El fin de los datos se indica con una línea que contiene sólo **</html>**.

Control 1 CC10A 2002 – Problema 3

La siguiente tabla muestra los métodos de una clase (Dia) que permite realizar operaciones con los días de una semana:

| Encabezamiento | ejemplo | resultado |
|---------------------------------|---------------------|---|
| Dia(int x) // $1 \leq x \leq 7$ | Dia a = new Dia(1); | objeto que representa el día lunes |
| Dia mañana() | a.mañana() | objeto que representa el día de mañana |
| int comparar(Dia x) | a.comparar(b) | 1 si los objetos son iguales (0 si distintos) |
| String nombre() | a.nombre() | “lunes” o “martes” o ... “domingo” |

A) Utilice la clase anterior en un programa que muestre el calendario de un mes siguiendo el diálogo:

```
Ingrese cantidad de días del mes (1-31) : 31
Ingrese primer día del mes (1-7) : 3
mi  ju  vi  sa  DO  lu  ma  mi  ju  vi  sa  DO  ...  ju  vi
1   2   3   4   5   6   7   8   9  10  11  12  ...  30  31
```

Nótese que:

- los nombres de los días aparecen abreviados y separados por un espacio
- los domingos se muestran en mayúsculas
- los números de los días aparecen bajo la primera letra del nombre del día

B) Escriba la clase Dia, suponiendo la declaración: `class Dia{ private int d; ... }`

Control 1 CC10A 2004 – Problema 2

a) Escriba un método (función) de encabezamiento `String traducir(String X,String Y,String Z)` que traduzca todos los caracteres de X que estén en Y, por los correspondientes caracteres de Z. Por ejemplo, `traducir(“Hola”,”aeiou”,”AEIOU”)` entrega “HOLA” y `traducir(“abracadabra”,”abcd”,”12222”)` entrega “1221212121”.

b) Los teléfonos celulares tienen un teclado donde las teclas numéricas se asocian a 3 o 4 letras:

| | | | | | | | | |
|--------|-----|-----|-----|-----|-----|------|-----|------|
| tecla | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| letras | ABC | DEF | GHI | JKL | MNO | PQRS | TUV | WXYZ |

Cuando un usuario escribe un mensaje debe presionar cada tecla varias veces para obtener la letra deseada. Por ejemplo, para la H debe presionar 2 veces la tecla 4. Por lo tanto, para escribir “HOLA” la secuencia es “446665552”. Para acelerar el proceso se utiliza escritura predictiva, que consiste en que el usuario escribe sólo 4652 y el teléfono revisa todas las palabras que se pueden escribir usando las letras asociadas a las teclas 4, 6, 5 y 2.

Al respecto, escriba un programa que lea un String numérico y escriba la primera palabra cuya escritura predictiva coincida con ese número. Por ejemplo, si lee “4652” escribe la palabra “HOLA”.

Notas.

- En el archivo “diccionario.txt” se encuentran todas las palabras posibles usando sólo las 26 letras que aparecen en el teléfono. Cada palabra está grabada en una línea del archivo.
- Para traducir cada palabra del archivo en su secuencia numérica **debe usar el método de la parte a)**

Control 1 CC10A 2004 – Problema 3

La siguiente tabla describe los métodos disponibles en la clase Perfil que permite manejar ternas de números reales entre 0 y 1:

| Ejemplos (Perfil p,q,r,) | Resultado | encabezamiento |
|--------------------------|--|--------------------------------------|
| p=new Perfil(x,y,z); | objeto que representa a la terna con valores x, y, z | Perfil(double x, double y, double z) |
| q=p.complemento(); | objeto con valores complementarios c/r a 1.0. Por ej, el complemento de (0.3,0.5,1.0) es (0.7,0.5,0.0) | Perfil complemento() |
| double d=p.distancia(q); | $ p_1-q_1 + p_2-q_2 + p_3-q_3 $ | double distancia(Perfil x) |
| p.mezclar(q); | Reemplazar los valores del objeto promediando sus valores con los valores del parámetro | void mezclar(Perfil x) |

A. (Ponderación: 1/3)

Escriba la clase Perfil

B. (Ponderación:2/3)

Escriba un programa **que use la clase anterior** para buscarle pareja a una persona cuyo perfil está descrito por valores entre 0 y 1 en los siguientes 3 aspectos: sentido del humor, romanticismo y sociabilidad. El criterio es elegir a la persona que esté a la menor distancia de las características opuestas o complementarias a la persona que busca pareja.

Notas.

- El programa debe establecer un diálogo como el que se muestra en el siguiente ejemplo:

```
Conteste las siguientes preguntas con valores entre 0 (nada) y
1 (mucho)
Sentido del humor ? 0.7
Romanticismo ? 0.3
Sociabilidad ? 1.0
Resultado: el nombre de su pareja es ....
```

- Las potenciales parejas se encuentran en el archivo “parejas.txt”. Cada persona en una línea con:
 - Nombre de 20 caracteres
 - 3 números reales entre 0 y 1 (escritos en la forma x.x) que corresponden a los valores asociados a los rasgos de humor, romanticismo y sociabilidad.
- Por ej, la línea “Knurthed Raimstedg W0.50.41.0” representa una persona con valores 0.5, 0.4 y 1.0.

Control 1 CC10A 2005 – Problema 2

Escriba un programa que le ayude al Papa recién designado a elegir su nombre como sumo pontífice, siguiendo un diálogo como el que se muestra en el siguiente ejemplo:

```
Ingrese nombre que desea ? Benedicto
Nombre del Papa = Benedicto XVI
```

Notas

- El archivo “Papas.txt” contiene la lista de todos los Papas ordenados cronológicamente. Cada línea contiene el nombre y el número romano (al final de la línea antecedido por un espacio). Por ejemplo, antes de la designación de Benedicto XVI, una de las líneas del archivo contenía “Benedicto XV”, y la última línea contenía “Juan Pablo II”.
- Puede usar (sin escribir) funciones para la conversión entre números romanos y decimales. Por ejemplo, romano(16) entrega “XVI” y decimal(“IX”) entrega 9.
- Si el nombre elegido por el nuevo Papa no se había usado previamente entonces debe seguirse del número romano I. Por ejemplo, “Francisco Javier I”.

Control 1 CC10A 2005 – Problema 3

La siguiente tabla describe los métodos disponibles en la clase Periodico que permite manejar números racionales en el intervalo [0,1[. Por ejemplo, 0.333.. tiene un período 3; 0.1666... tiene un período 6 y 0.11234234234... tiene un período 234.

| Ejemplo (Periodico p;) | Resultado | Encabezamiento |
|-------------------------------|--|------------------------|
| p=new Periodico(0.11234234) | Objeto con racional 0.11234... | Periodico(double x) |
| p=new Periodico(“0.11234234”) | Objeto con racional 0.11234... | Periodico(String x) |
| p=new Periodico(11223,999000) | Objeto con racional 0.11234... que resulta de 11223/999000 | Periodico(int x,int y) |
| p.toString(9) | Entrega “0.1123423” (9 caracteres) | String toString(int x) |
| p.numerador() | numerador del racional p | int numerador() |
| p.denominador() | denominador del racional p | int denominador() |

a)Escriba un programa que use la clase anterior para mostrar $1/n$ (para n múltiplo de 3) con 1, 2, 3, ..., y 9 decimales en la forma:

| 1/n | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|------|-----|------|-------|--------|---------|----------|-----------|------------|-------------|
| 1/3 | 0.3 | 0.33 | 0.333 | 0.3333 | 0.33333 | 0.333333 | 0.3333333 | 0.33333333 | 0.333333333 |
| 1/6 | 0.1 | 0.16 | 0.166 | 0.1666 | 0.16666 | 0.166666 | 0.1666666 | 0.16666666 | 0.166666666 |
| 1/9 | 0.1 | 0.11 | 0.111 | ... | | | | | |
| ... | | | | | | | | | |
| 1/99 | 0.0 | 0.01 | 0.010 | 0.0101 | 0.01010 | ... | | | |

Nota. No es necesario que escriba las separaciones entre filas y columnas.

b) escriba los métodos **toString**, **numerador** y **denominador** de la **clase Periodico**, suponiendo que los objetos de la clase se representan por 4 números enteros:

- período (y)
- número antes del período (x)
- N° de dígitos del período (ny)
- N° de dígitos antes del período (nx).

Por ejemplo, el 0.11234234...se representa por $x=11$, $y=234$, $nx=2$, $ny=3$.

| | | | |
|---|-----|----|---|
| x | 11 | nx | 2 |
| y | 234 | ny | 3 |

Nota.

el número 0,11234234... se expresa como fracción en la forma

$$\frac{999*11+234}{999*100}$$

En que 999 es $10^{ny}-1$ y 100 es 10^{nx}

Control 1 CC10A 2006 – Problema 2

Prontamente, las nuevas patentes comenzarán con 4 letras y terminarán con 2 dígitos (por ejemplo ABCD12). Al respecto:

a) escriba una función de encabezamiento

```
static public boolean validar (String patente)
```

que entregue true si la patente está correcta (o false si no). Una patente se considera correcta si comienza con 4 letras y termina con 2 dígitos. Es importante señalar que no se permiten las letras I, M, Ñ, O, Q y W por la posible confusión visual.

Por otra parte, no se permiten patentes que en sus 4 primeras letras formen una palabra ofensiva. Al respecto, las palabras ofensivas están grabadas en el archivo “ofensivas.txt”. El archivo está ordenado alfabéticamente y cada palabra está grabada en una línea.

b) escriba un programa que escriba la lista de todas las patentes válidas en un cierto rango siguiendo el diálogo indicado en el siguiente ejemplo:

patente inicial? ABCD98 patente final? ABCF23

lista de patentes: 1:ABCD98 2:ABCD99 3:ABCE01 ... n°:ABCF23

Nota.

- utilice la función validar para comprobar que las patentes inicial y final están correctas. Si alguna está incorrecta termine el programa con el mensaje “patente incorrecta”
- suponga que existe una función que recibe una patente y entrega la siguiente patente válida. Por ejemplo, siguiente(“ABCD99”) entrega “ABCE01”.

Control 1 CC10A 2006 – Problema 3

La siguiente tabla describe los métodos disponibles en la clase Rectangulo:

| Ejemplos | Resultado | Encabezamiento |
|------------------------|--|-----------------------------|
| a=new Rectangulo(4, 3) | crea un objeto de clase Rectangulo (de ancho 4 y alto 3) | Rectangulo (int x, int y) |
| a.dibujar(“*”) | dibuja Rectangulo en la forma: * * * * 1ª línea: 4 asteriscos (ancho es 4) * * * * 2ª línea: 1 *, 2 espacios y 1 * * * * * 3ª línea: 4 asteriscos | void dibujar(char x) |
| a.compareTo(b) | 0 si el área de a es igual al área de b n°<0 si a<b (área de a < área de b) n°>0 si a>b (área de a > área de b) | int compareTo(Rectangulo x) |

a)Escriba la clase Rectangulo

b) Escriba la clase Rectangulo1, que extienda la clase Rectangulo, con los siguientes métodos

| Ejemplos | Resultado | Encabezamiento |
|-------------------------|--|---------------------------|
| c=new Rectangulo1(5, 4) | Crea un objeto de clase Rectangulo1 (de ancho 5 y alto 4) | Rectangulo1(int x, int y) |
| c.pintar(“*”) | pinta rectángulo en la forma: * | void pintar(char x) |
| c.rotar() | Entrega un nuevo rectángulo, rotando el original en 90 grados, es decir de ancho 4 y alto 5. | Rectangulo1 rotar() |

Control 2 CC10A 2001 – Problema 1

La siguiente clase permite asignar y recuperar los valores de una terna de números enteros:

```
class Terna
{
    int a,b,c;

    public Terna()
    { a=0; b=0; c=0; }

    public void set1(int x)
    { a=x; }

    public void set2(int x)
    { b=x; }

    public void set3(int x)
    { c=x; }

    public int get1()
    { return a; }

    public int get2()
    { return b; }

    public int get3()
    { return c; }
}
```

- a. Escriba la clase **Triangulo** que extienda la clase **Terna** agregando los siguientes métodos:

| ejemplo | resultado | significado |
|------------------|------------------|---|
| Triangulo() | - | constructor que inicializa con ceros los valores de los tres lados. |
| T.esTriangulo() | boolean | true si T corresponde a un triangulo o false si no (3 números positivos forman un triángulo si todas las sumas de 2 de ellos son mayores que el 3°) |
| T.ladosIguales() | int | cantidad de lados iguales (0, 2 o 3) del triángulo T |
| T.graficar(x) | void | grafica el triángulo T de color x (String "rojo", "azul" o "verde") |

Nota. NO debe escribir el método graficar

- b. Escriba un programa que dibuje 100 triángulos (de lados enteros entre 1 y 100 generados al azar) de modo que los equiláteros (tres lados iguales) se dibujen de color rojo, los isósceles (dos lados iguales) de color azul, y los escalenos (todos los lados distintos) de color verde.

Nota. Recuerde que `Math.random()` devuelve un N° real de tipo double en el rango `[0,1[`

Control 2 CC10A 2001 – Problema 2

La siguiente figura muestra la interfaz de un programa (Frame) que permite apoyar el recuento de votos de una elección con dos candidatos ("Rosa María" y "José Manuel"):

| | | | |
|--------|-----------------------|-------------|-------------------|
| Label | Candidato | N° de Votos | Label |
| Button | Rosa María | 0 | TextField o Label |
| Button | José Manuel | 0 | TextField o Label |
| Label | Total | 0 | TextField o Label |
| Button | descontar último voto | recomenzar | Button |

Escriba la clase que controle la interfaz anterior de acuerdo a las siguientes reglas:

- Cada vez que se da un click en el botón de un candidato se debe incrementar en uno el N° de votos del candidato (y el total de votos)
- El botón "recomenzar" debe volver a poner ceros en los tres contadores de votos
- El botón "descontar último voto" sirve para corregir un error al registrar la última preferencia, es decir, debe descontar un voto al candidato que recibió el voto más reciente (y al total de votos)

Control 2 CC10A 2002 – Problema 1

Para mantener una pieza de Ajedrez en el tablero de 8 filas y 8 columnas, se dispone de la clase:

```
class Pieza{
    private int fila, columna; //private puede cambiarse por protected u omitirse
    public Pieza( int x,int y){ fila=x; columna=y; }
    public boolean mover(int x,int y){fila=x; columna=y; return true;}
    public int obtenerFila(){return fila;}
    public int obtenerColumna(){return columna;}
}
```

A) (**ponderación: 2/3**) Escriba las clases Caballo y Torre que extiendan la clase Pieza con los métodos:

| ejemplo | significado |
|------------------------------|---|
| C=new Caballo(1,1) | constructor que ubica un caballo en fila 1 y columna 1 |
| C.mover(2,3) C.mover(1,2) | Mueve C a fila 2 y col. 3, y entrega true (movimiento válido) Entrega false y no mueve el caballo (movimiento inválido). Nota. Un movimiento de un caballo es válido si avanza 2 filas y 1 columna o 2 columnas y 1 fila dentro del tablero. |
| T = new Torre(5,6) | constructor que ubica una torre en fila 5 y columna 6 |
| T.mover(5,8) T.mover(5,9) | Mueve T a fila 5 y col 8, y entrega true (movimiento válido). Entrega false y no mueve la torre (movimiento inválido). Nota. Es válido si avanza en la misma fila o columna dentro del tablero. |

B) (**ponderación: 1/3**) Escriba el método **boolean comer(Pieza X,Pieza Y)** que devuelva true si la Pieza X “come” en una jugada a la Pieza Y, considerando que no hay más piezas en el tablero. Por ejemplo, si **Caballo C=new Caballo(1,2)** y **Torre T=new Torre(3,1)**, entonces **comer(C,T)** devuelve true y **comer(T,C)** devuelve false.

Control 2 CC10A 2002 – Problema 2

La siguiente figura muestra la interfaz de un Frame que permite ingresar una palabra en código morse en la forma indicada en el siguiente ejemplo en que se ingresa la palabra OLA.

| | | | |
|--------|------------------------|-----------------------|--------|
| Button | punto | raya | Button |
| Button | espacio | ingresar otra palabra | Button |
| Label | Palabra en morse: | - . - . - . - . - . - | Label |
| Label | Palabra en castellano: | OLA | Label |

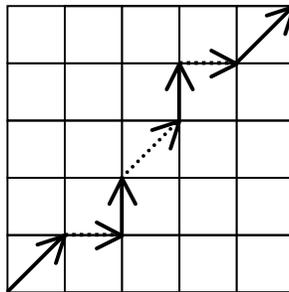
Escriba la clase que controle la interfaz anterior de acuerdo a las siguientes reglas:

- un click en el botón “punto” o “raya” debe ingresar un carácter . o un carácter -
- un click en el botón “espacio” indica fin del código de una letra
- un click en el botón “ingresar otra palabra” restablece la situación inicial

Nota. Suponga que existe el método Morse que devuelve la letra correspondiente a un código. Por ejemplo, Morse(“.-..”) devuelve “L”.

Control 2 CC10A 2003 – Problema 2

Un juego llamado "carrera de flechas" se juega entre dos personas sobre un papel cuadrado en el cual alternadamente los jugadores van dibujando una flecha desde un punto al siguiente más próximo ya sea hacia la derecha, hacia arriba o en diagonal (arriba derecha). Cada jugador debe dibujar su flecha desde el punto donde el jugador anterior terminó de dibujar la suya. El jugador que parte lo hace dibujando una flecha desde el extremo inferior izquierdo. Gana el jugador que dibuja la flecha que llega al extremo superior derecho del cuadrado.



En este ejemplo, el cuadrado es de 5x5, empezó el jugador con línea llena y el mismo jugador ganó el juego

a) (ponderación: 1/3) La siguiente clase permite manipular y dibujar flechas:

```
public class Flecha {
    protected int x0, y0; //coordenadas de origen de la flecha
    public Flecha (int x, int y) { x0 = x; y0 = y; }
    public void dibujar(int x, int y, G g) { . . . }
    // G = Graphics
}
```

Complete el método dibujar, de modo que trace una línea desde el origen (x0,y0) hasta (x, y) y cambiar el origen a (x, y). Nota: no necesita dibujar la punta de la flecha.

b) (Ponderación 2/3) Escriba un programa que utilizando la clase anterior, implemente una interfaz gráfica para jugar "carrera de flechas" entre dos personas en un cuadrado de 10x10 cuadrados de 10x10 pixeles cada uno. Además debe proveer tres botones con las leyendas "horizontal", "vertical" y "diagonal" que permiten al jugador de turno dibujar la flecha que estime conveniente. El programa debe terminar cuando algún jugador llega al extremo superior izquierdo.

Nota: El cuadrado y los botones pueden aparecer en la misma o en distintas ventanas, y en cualquier ubicación.

Control 2 CC10A 2004 – Problema 1

La siguiente clase permite manejar números enteros positivos muy grandes (con muchos dígitos y sin ceros a la izquierda) imposibles de representar en un int:

```
class Numero{
protected String digitos;//la palabra protected se puede omitir
public Numero(String x){ digitos=x; }
public String toString(){ return digitos; }
public int compareTo(Numero x){
    if(digitos.length()x.digitos.length()) return 1; //positivo
    return digitos.compareTo(x.digitos);//Nº cero, positivo o
negativo
}
}
```

- Escriba la clase ConSigno, que extienda la clase Numero, para manejar números enteros muy grandes con signo (+ o -) y sin ceros a la izquierda.
- Utilice la clase anterior en un programa que muestre el menor número grabado en el archivo “enterosgrandes.txt”. El archivo contiene una cantidad indeterminada de líneas, cada una con un N° entero de muchos dígitos (sin ceros a la izquierda y con un signo + o - antes del primer dígito).

Control 2 CC10A 2004 – Problema 2

Escriba un programa que dibuje una sólo vez la posición inicial de un tablero de NxN casilleros (N par > 2) de un juego de reversi (othello), en que las fichas de los jugadores se pinten **Yellow/Black** o **Red/Blue**. El programa debe establecer la interfaz con el usuario que se muestra en el siguiente ejemplo con N=6 y el botón Y/B:

| Label | TextF | Button | Button | Button |
|-------|-------|--------|--------|--------|
| N? | 6 | Y/B | R/B | quit |
| | | | | |
| | ○ | ● | | |
| | ● | ○ | | |
| | | | | |
| | | | | |

- Las 4 fichas centrales se dibujan como **círculos o rectángulos** de cualquier tamaño(dentro del casillero)
- El dibujo debe mostrarse al momento del click en el botón que selecciona el color de las fichas

Control 2 CC10A 2005 – Problema 1

a) Escriba la clase Mechon que permita manejar las notas de los dos controles del primer semestre de un alumno de CC10A a través de los métodos indicados en la siguiente tabla:

| Ejemplo (Mechon a;) | Significado |
|----------------------------|---|
| a=new Mechon("Ana"); | crea objeto de clase Mechón que representa a una mechona de nombre Ana |
| a.notaC1(4.8); | Asigna la nota 4.8 en el control1 |
| a.notaC2(5.3); | Asigna la nota 5.3 en el control2 |
| a.promedio() | Entrega el promedio en los dos controles(si no rindió alguno se considera una nota 1.0) |

b) Escriba la clase Mechon1, que extienda la clase anterior, de modo que mantenga también la nota del control recuperativo y la maneje a través de los siguientes métodos:

| Ej(Mechon1 b;) | significado |
|-----------------------|--|
| b.notaCR(6.2); | Asigna la nota 6.2 en el control recuperativo |
| b.promedio() | Calcula el promedio en los dos controles, considerando que la nota del control recuperativo reemplaza a la menor de las notas de los controles (salvo que la nota del recuperativo sea peor) |

c) (No es aplicable al Control 2 2008)

Control 2 CC10A 2005 – Problema 2

La siguiente figura muestra la forma inicial de la interfaz con el usuario de un programa que permite responder una prueba con preguntas verdadero o falso.

| | | | |
|-------------------|---|-------------|--------|
| Label o TextField | Pregunta 1: La capital de Brasil es Sao Paulo | | |
| Button | Verdadero | Falso | Button |
| Button | Cara (V) o sello (F) | No contesta | Button |
| Label o TextField | Puntaje acumulado=0.0 | quit | Button |

Escriba el programa completo que controle la interfaz anterior de acuerdo a las siguientes reglas:

- un click en el botón quit debe terminar el programa
- un click en el botón “Verdadero” o en el botón “Falso” representa la respuesta a la pregunta
- un click en el botón “cara o sello” representa una respuesta V o F elegida al azar (random)
- un click en el botón “no contesta” significa dejar la pregunta sin contestar
- un click en cualquier botón causa que el puntaje acumulado se actualice y se muestre la siguiente pregunta de la prueba (precedida por el N° de la pregunta).

- el puntaje acumulado debe incrementarse en uno si la respuesta es correcta, debe disminuirse en 0.5 si la respuesta es incorrecta, y mantenerse en caso de no contestar
- cada línea del archivo “test.txt” contiene una pregunta de la prueba, con la respuesta correcta (V o F) en el último carácter. Por ej, “La capital de Brasil es Sao PauloF”, indica que la respuesta correcta es Falso.
- después de responder la última pregunta, debe mostrarse el texto “no hay más preguntas”, sin actualizar el puntaje acumulado

Control 2 CC10A 2006 – Problema 1

Considerando la siguiente interfaz:

```
interface I{
public void leer(String x);
public String escribir();
public int perimetro();
}
```

- a) Escriba las clases Rectangulo y Triangulo que implementen la *interface* I para representar, calcular el perímetro, leer y escribir las figuras correspondientes.

Notas.

- Las figuras se representan por las longitudes de sus lados (2 valores enteros para rectángulos y 3 valores para triángulos)
- El método leer, debe obtener desde el string los valores (separados por espacios) de los lados de la figura (2 o 3 valores enteros según corresponda)
- El método escribir debe entregar un string con los valores de los lados separados por espacios (2 o 3 valores enteros)

- b) Escriba un programa que lea el archivo “figuras.txt” que en cada línea contiene una letra (R o T) y a continuación 2 o 3 números (valores de los lados separados por espacios). El programa debe usar las clases anteriores para escribir como resultado los valores de los lados de la figura de mayor perímetro. Por ejemplo, si el archivo contiene las líneas:

```
R 10 20
T 30 40 5
R 20 30
T 1 2 3
```

entonces el resultado es 20 30, puesto que 100 es el mayor perímetro y corresponde al rectángulo de lados 20 y 30.

Control 2 CC10A 2006 – Problema 2

La figura ubicada a la derecha ilustra una interfaz gráfica que permite dibujar árboles. El dibujo de un árbol está compuesto de una rama inicial (tronco) que se divide en dos ramas separadas por un ángulo y que miden $\frac{2}{3}$ de la longitud de la rama inicial.

Escriba un programa que implementa la interfaz ilustrada. Los dos campos de texto (objetos de tipo TextField) permiten determinar los parámetros del dibujo: la profundidad de ramas en que se abre el árbol y el ángulo en que se abren las ramas.

Notas

- El botón etiquetado “Dibujar” dibuja el árbol teniendo en cuenta los dos parámetros indicados en los campos de texto (que tienen valores iniciales 10 y 30)
- Utilice un BorderLayout, con un Canvas al centro y paneles al norte(2 filas y 2 cols) y sur(1 fila y 2 cols)
- El botón cerrar termina la ejecución del programa.
- Se recomienda usar y escribir un método recursivo de encabezamiento:

```
void dibujarArbol(int profundidad, int largoTronco, double angulo,
int x, int y, Graphics g) //coordenadas iniciales (horizontal y vertical) del
tronco en el objeto g
```

