Guía de Estudio

Control 1

Juan Alvarez – Nelson Baloian

Kurt Schwarze – Eric Reimberg – Andrés Muñoz

CC10A - 2003

Tabla de Contenidos

1. Promedio de Notas (Pregunta 2 Control 1, 1994)

Unos profesores usan la siguiente política en sus ramos: "*La nota final se calcula promediando todas las notas, pero reemplazando la peor por la mejor*" (o sea se elimina la peor y la mejor vale por dos).

Escriba un programa que lea una lista de notas reales (terminadas por un 0), e imprima el promedio según esta política.

Solución:

```
class Peor {
      static public void main (String args[]) {
            // Declaraciones necesarias
            Console c = new Console();

            double mejor = 1;
            double peor = 7;
            double notas = 0;
            double suma = 0;
            int cont = 0;

            // Ingreso de notas
            c.println("Ingrese las notas");
            c.print("?");
            while((notas = c.readDouble())!=0) {
                  ++cont;
                  if (peor>notas)
                        peor = notas;
                  if (mejor<notas)
                        mejor = notas;
                  suma+=notas;
                  c.print("?");
            }

            // Reemplazo de la peor nota
            suma = suma - peor + mejor;

            // Imprime el promedio
            c.println("El promedio es "+suma/cont);
      }
}
```


2. Digito Verificador (Pregunta 3 Control 1, 1994)

Se quiere hacer un programa que calcule el dígito verificador del RUT. La idea es que se tiene un número entero que es el RUT (con un número de dígitos desconocido y sin dígito verificador) en una variable rut.

El cálculo del dígito verificador se hace tomando los dígitos del entero uno por uno, se multiplican por una constante asociada a la posición y el resultado se va sumando a un total. Finalmente, el total es dividido por 11 y se toma el resto r. El dígito es 11 - r. Si es igual a 11 se usa el "0", si es igual a 10 se usa "k".

El método parte del dígito menos significativo y sigue hasta el último (ver figura 1).

1	3	.	2	5	2	.	3	1	1
d8	d7		d6	d5	d4		d3	d2	d1

p = d1*2 + d2*3 + d3*4 + d4*5 + d5*6 + d6*7 + d7*2 + d8*3 + ...

digito = 11 - (p mod 11)

si digito = 10 => k
si digito = 11 => 0

Ej.: rut = 13252311

p = 1*2 + 1*3 + 3*4 + 2*5 + 5*6 + 2*7 + 3*2 + 1*3 = 80

digito = 11 - (80 mod 11) = 11 - 3 = 8

Solución:

```
class Digito {
      static public void main (String args[]) {
            // Declaraciones
            Console c = new Console();

            int suma = 0;
            int digito = 0;
            int j = 1;

            // Ingreso del RUT
            c.println("Ingrese el RUT");
            String rut = c.readLine();
            int carnet = Integer.parseInt(rut);

            // Calculo de la suma ponderada
            for (int i=rut.length();i>=0;--i) {
                  ++j;
                  if (j>7) j=2;
                  suma += (carnet % 10) * j;
                  carnet = carnet / 10;
            }
```

```
            // Calculo del valor del digito
            digito = 11 - suma % 11;

            // Imprimir el digito
            if (digito == 11)
                  c.println ("El digito verificador es 0");
            else if (digito == 10)
                  c.println ("El digito verificador es k");
            else
                  c.println ("El digito verificador es "+digito);
      }
}
```


3. Separador de Palabras (Pregunta 3 Control 2, 1994)

Se pide leer de la pantalla 2 strings: uno con una lista de caracteres considerados *Separadores* y otro que es una línea de texto cualquiera.

El programa debe generar un arreglo de strings, con todas las palabras contenidas en la línea de texto, sin los separadores. Finalmente, debe imprimir este arreglo, y el número de palabras que hay en él. Suponga que hay una variable MAXPAL que indica el máximo de palabras en cada línea de texto.

Una secuencia de separadores es lo mismo que tener uno solo.

Ejemplo:

```
Separadores? ,. ;
      Linea? ,hola... como te va?
Palabras:
      hola
      como
      te
      va?
(4 palabras)
```

Solución:

```
class Tokenizador {
      static public void main (String args[]) {
            // Declaraciones
            Console c = new Console();

            String seps;
            String linea;
            int contador=0, indice=0, inicio=0;
            bolean sep = false;

            // Ingreso del usuario
            c.print("Separadores? ");
            seps = c.readLine();
            c.print("Linea? ");
            linea = c.readLine();

            // Busqueda de separadores en la linea
            while (indice <= linea.length()) {
                  if (seps.indexOf(linea.chartAt(indice))>0) {
                        if (indice > inicio)
                              c.println("     " + palabra);
                        inicio = indice + 1;
                        sep = true;
                        palabra = "";
                  }
                  else {
                        palabra = palabra + linea.charAt(indice);
                  }
                  indice++;
            }
            c.println("("+contador+" palabras)");
      }
}
```

4. Niveles de Contaminación (Pregunta 1 Control 1, 1997)

En la entrada de Avda. Tupper del Parque O'Higgins se ubica la estación que presenta siempre los niveles más altos de contaminación en Santiago. Al respecto, escriba un programa que permita realizar las siguientes operaciones con la información del índice de contaminación:

Oper.	Significado
1	Solicitar y guardar el valor del índice y la hora en que fue medido
2	Mostrar el último valor del índice y la hora en que se registró
3	Mostrar el valor máximo del índice hasta ahora medido y la hora en la cual se produjo
4	Mostrar el promedio de los valores obtenidos para el índice
0	Fin de las operaciones

Notas:

En el caso que el índice alcance el valor 400 escriba el mensaje "Emergencia". Posteriormente, la primera vez que el índice registre un valor menor que 400 escriba "Fin Emergencia".
El siguiente diálogo muestra en ejemplo de las operaciones:

```
Operacion? 1
Indice y Hora? 200 1030

Operacion? 1
Indice y Hora? 420 1115
Emergencia

Operacion? 1
Indice y Hora? 280 1210
Fin Emergencia

Operacion? 2
Indice=280   Hora=1210

Operacion? 4
Promedio Indice=300

Operacion? 3
Maximo Indice=420   Hora=1115

Operacion? 0
Fin Operaciones
```

Solución:

```
class Estacion {
      static public void main (String args[]) {
            // Declaraciones
            Console c = new Console();
            int oper = 10;  // Solo como valor inicial

            int ind_max=0, hor_max=0;
            int ind_ult=0, hor_ult=0;
            int ind_sum=0, tot_ind=0;
            boolean emergencia=false;
```



```
            // Ciclo de operaciones
            while (oper>0) {
                  c.print("Operacion? ");
                  oper = c.readInt();

                  // Distintos casos
                  switch (oper) {
                  case 0:
                        break;
                  case 1:
                        c.print("Indice y Hora? ");
                        ind_ult = c.readInt();
                        hor_ult = c.readInt();

                        // Vemos si hay maximo
                        if (ind_max<ind_ult) {
                              ind_max=ind_ult;
                              hor_max=hor_ult;
                        }

                        // Sumamos a los totales
                        ind_sum += ind_ult;
                        ++tot_ind;

                        // Vemos si hay emergencia
                        if (ind_ult>=400 && !emergencia) {
                              c.println("Emergencia");
                              emergencia=true;
                        }
                        else if (ind_ult<400 && emergencia) {
                              c.println("Fin Emergencia");
                              emergencia=false;
                        }
                        break;
                  case 2:
                        // Imprimimos ultimos datos
                        c.print("Indice="+ind_ult+
                                " Hora="+hor_ult);
                        break;
                  case 3:
                        // Imprimimos datos maximos
                        c.print("Indice="+ind_max+
                                " Hora="+hor_max);
                        break;
                  case 4:
                        // Imprimimos promedio
                        c.print("Indice="+ind_sum/tot_ind);
                        break;
                  default:
                        c.println ("Error de Operacion");
                  }
            }
            c.println("Fin Operaciones");
      }
}
```

5. Cachipún (Problema 1 Control 1, 1998)

Un alumno de la escuela de ingenieria quiere tene un programa para entretenerse cuando este frente al computador pero no tenga nada que hacer. Asi es como se le ocurrio que podia escribir un programa sencillo para jugar al cachipun contra el computador. El dialogo que debe tener el jugador con la maquina debe ser el siguiente:

```
Significado: 1-papel, 2-tijeras, 3-piedra
ca-chi-pun? 2
computador juega 3
Cuenta: Persona 0 Computador 1
ca-chi-pun? 2
computador juega 1
Cuenta: Persona 1 Computador 1
ca-chi-pun? 3
computador juega 3
Cuenta: Persona 1 Computador 1
```

Solución

```
class Cachipun {
      // Un método que calcule un número aleatorio genérico
      static int aleatorio (int min, int max) {
            double r = Math.random();
            return Math.trunc(r * (max – min + 1)) + min;
      }

      static public void main (String[] args) {
            Console c = new Console();
            c.println("Significado: 1-papel, 2-tijeras, 3-piedra");

            // Inicialización de variables de juego
            int jugador = 0;
            int computador = 0;
            int contadorJugador = 0, contadorComputador = 0;

            // Ciclo infinito
            while (true) {
                  c.println("ca-chi-pun? ");
                  jugador = c.readInt();

                  computador = aleatorio(1, 3);
                  c.println("computador juega " + computador);

                  if ( (jugador == 1 && computador == 3)
                              || jugador > computador)
                        contadorJugador ++;
                  else if ( (jugador == 3 && computador == 1)
                              || jugador < computador)
                        contadorComputador ++;

                  // Resultado Acumulado
                  c.print("Cuenta");
                  c.print("Persona " + contadorJugador);
                  c.print("Computador " + contadorComputador);
            }
      }
}
```


6. Distancia 2 Puntos (Problema 1 Control 2, 1998)

a) La distancia entre dos puntos de coordenadas cartesianas (x1,y1) y (x2,y2) se define como:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Escriba un método llamado distancia que reciba las coordenadas (reales) de 2 puntos y devuelva su distancia.

b) El algoritmo de Montecarlo permite obtener una aproximación de PI de la siguiente manera:

- Considerar un circulo inscrito en un cuadrado de lado 2
- Generar al azar varios puntos (x,y) dentro del cuadrado (usando Math.random())
- Contar el número de puntos que caen dentro del círculo (cuyo centro está en (1,1))

Aproximar PI considerando la proporción de entre los puntos que "cayeron" dentro del círculo y el total de puntos, y la proporción entre las áreas del círculo y el cuadrado.

Solución

a)

```
public double distancia (double x1, double y1, double x2, double y2)
{
      double d;
      d = Math.sqrt(Math.pow(x2 - x1, 2) + Math.pow(y2 - y1, 2));
      return d;
}
```

b)

```
Console c = new Console();
double pi = 0;
double ndentro = 0, npuntos = 1000;
for (int i=0; i<npuntos; i++) {
      double x = Math.random() * 2;
      double y = Math.random() * 2;
      npuntos++;
      if (distancia(1, 1, x, y) <= 1) { // DENTRO
            ndentro++;
      }
}
pi = ndentro / npuntos;
c.println("PI = " + pi);
```

7. Declaración de Impuestos (Pregunta 1 Control 1, 2000)

Escriba un programa que calcule el monto de impuestos que debe pagar una persona, ciñéndose al diálogo que se muestra en el siguiente ejemplo:

```
Mes 1
Ingresos Percibidos en $ ? 300000
Factor de Actualizacion ? 1.020
Ingresos Actualizados = $ 306000

...

Mes 12
Ingresos Percibidos en $ ? 250000
Factor de Actualizacion ? 1.000
Ingresos Actualizados = $ 250000

Suma de ingresos actualizados = $ xxxxxxxxxx
Impuestos a pagar = $ xxxxxx
```

Nota:
- Los ingresos actualizados de cada mes se obtienen aplicando el factor a los ingresos percibidos.
- Para calcular los impuestos a pagar deben aplicar las siguientes reglas (UTM = $15000)

§ Si Total > 10 UTM => Exento de Impuestos
§ Si Total >= 10 UTM y < 30 UTM => 5% de Impuestos
§ Si Total >= 30 UTM y < 50 UTM => 10% de Impuestos
§ Si Total >= 50 UTM => 15% de Impuestos

Solución

```
class Impuestos {
      public static void main (String args[]) {
            Console c = new Console() ;
            int suma = 0 ;
            for ( int i = 1 ; i <= 12 ; ++i ) {
                  c.println("Mes "+i) ;
                  c.print("Ingresos percibidos en $ ? ") ;
                  int ingresos = c.readInt() ;
                  c.print("Factor de actualizacion ? ") ;
                  double factor = c.readDouble() ;
                  suma = suma + (int) (ingresos * factor) ;
              }
            int impuestos = 0 ;
            int utm = 15000 ;
            if ( suma < 10 * utm )
                  impuestos = 0 ;
            else if ( suma < 30 * utm )
                  impuestos = (int)(suma * 0.05) ;
            else if ( suma < 50 * utm )
                  impuestos = (int)(suma * 0.10) ;
            else
                  impuestos = (int)(suma * 0.15) ;
            c.println("Suma de ingresos actualizados = $ "+suma) ;
            c.println("Impuestos a pagar = $ "+impuestos);
        }
  }
```


8. Subrayado de Palabras (Pregunta 2 Control 1, 2000)

Escribir un programa que lea una linea (o que la reciba por parametro un metodo DECIDE TU) y que le ponga un subrayado debajo de la palabra que comienza con un &, es decir:

```
Ingrese frase:
? Se que me ira &bien en el control de &Computacion
Frase subrayada:
> Se que me ira bien en el control de Computacion
                ----                  -----------
```

Nota: La palabra subrayada comienza con un & y termina con un ESPACIO. Los & no aparecen en el resultado.

Solución

```
class Subrayado {
      public static void main (String args[]) {
            Console c = new Console() ;
            c.print("? ") ;
            String linea = c.readLine() ;
            boolean sub = false ;
            String linea1 = "", linea2 = "" ;

            for ( int i = 0 ; i < linea.length() ; ++i ) {
                  char d = linea.charAt(i) ;
                  if ( sub ) {
                        if ( d != ' ' ) {
                              linea1 += d ;
                              linea2 += "-" ;
                          }
                        else {
                              linea1 += d ;
                              linea2 += " " ;
                              sub = false ;
                          }
                    }
                  else {
                        if ( d != '&' ) {
                              linea1 += d ;
                              linea2 += " " ;
                          }
                        else {
                              sub = true ;
                          }
                    }
              }
            c.println("> "+linea1) ;
            c.println("  "+linea2) ;
        }
  }
```

9. Competitividad de Paises (Pregunta 3 Control 1, 2000)

Recientemente se publicó un informe de competitividad mundial de los paises en que Chile aparece en el lugar 26 (año 2000, pero ahora aparece 24... GUAU). Para determinar el lugar que ocupa un país se calcula un puntaje de acuerdo a la evaluacion (buena, regular, mala) en 50 aspectos distintos (ej: gasto en salud, educacion, etc).

Al respecto escriba un programa que escriba los nombres de los países que ocuparon el primer y segundo lugar en el ranking. Los dados se deben obtener del archivo "paises.txt", que en cada línea contiene la siguiente información de un país:

- Nombre del país (20 primeros caracteres)
- Evaluación en cada uno de los 50 aspectos (como B, R o M).

Por ejemplo:

```
Australia           BRBMMMRRBMRRMBBRMMBRRMMMBBBBRMBRMMBRMBRBMRMBBMRMBR
Alemania            BMRMMBBBBBBBMBMBMBMRRRRRRBMMRMBRMBMRBMRMBMRMBRMBMR
...
```

Nota: Para calcular el puntaje de un país se deben sumar las evaluaciones en cada aspecto. Para ello considere que (B)ueno, (R)egular y (M)alo equivale a valores 3, 2 y 1 respectivamente.

Solución

Para aplicar esta solución, cabe explicar que, leer una línea desde un archivo es muy similar a leerla desde el teclado. Para ello utilizas la clase BufferedReader y la clase FileReader de la siguiente forma:

1. Abres un archivo para leer su contenido:

```
BufferedReader file = new BufferedReader(
      new FileReader("<nombre del archivo>")) ;
```

2. Lees una línea desde el archivo:

```
String linea = file.readLine();
```

Con estas 2 líneas, quedamos con la siguiente solución:

```
import java.io.* ;
class Ranking {
      public static void main (String args[]) throws IOException {
            Console c = new Console() ;
            BufferedReader file = new BufferedReader(
                  new FileReader("paises.txt")) ;
            String linea ;
            String pais1 = "", pais2 = "" ;
            int nota1 = 0, nota2 = 0 ;

            while ( (linea = file.readLine()) != null ) {
                  String pais = linea.substring(0,20) ;
                  String eval = linea.substring(20) ;
```



```
                  int nota = 0 ;
                  for ( int i = 0 ; i < eval.length() ; ++i ) {
                        char d = eval.charAt(i) ;
                        if (d == 'B')
                              nota += 3 ;
                        else if (d == 'R')
                              nota += 2 ;
                        else
                              nota += 1 ;
                    }
                  if ( nota > nota1 )  {
                        nota2 = nota1 ;
                        pais2 = pais1 ;
                        nota1 = nota ;
                        pais1 = pais ;
                    }
                  else if ( nota > nota2 ) {
                        nota2 = nota ;
                        pais2 = pais ;
                    }
              }

            c.println("Primer lugar : "+pais1) ;
            c.println("Segundo lugar: "+pais2) ;
      }
}
```

10. Números Pitagóricos (Pregunta 1 Control 1, 2001)

Tres números enteros positivos a, b y c se consideran pitágoricos si cumplen con la relación

$$a^2 + b^2 = c^2$$

Por ejemplo 3, 4 y 5 son pitágoricos puesto que 32+42=52. Nótese que si los números representan las longitudes de los lados de un triángulo, entonces forman un triángulo rectángulo de catetos a y b e hipotenusa c.

Al respecto, escriba un programa que lea el valor de la hipotenusa y busque los valores de los catetos del triángulo rectángulo correspondiente, siguiendo el diálogo indicado en el siguiente ejemplo:

```
Determinar catetos de triangulo rectangulo
Ingrese valor de hipotenusa: 10
Catetos = 6 8
```

En caso que no se puedan obtener valores enteros para los catetos debe responderse "Catetos no son enteros".

Solución

```
Console C=new Console();

// obtener hipotenusa
C.println("Determinar catetos de triangulo rectangulo")
C.println("Ingrese valor de hipotenusa")
int c= C.readInt();

// inicizalizar indicador
boolean sonEnteros = false;  // tambien puede ser de tipo int

// iterar con valores enteros
for (int a=1; a < c; ++a) {
      for (int b=1;b < c; ++b)
            // consultar si son catetos
            if(a*a + b*b == c*c){

            // mostrar valores de catetos
            C.println("catetos="+ a + " y " + b);

            // indicar que se encontraron catetos enteros
            sonEnteros = true;
            break;
      }
}
// mostar mensaje si catetos no son enteros
if( sonEnteros == false)
C.println("catetos no son enteros");
```


11. Eliminador de Strings (Pregunta 2 Control 1, 2001)

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
```

Solución

a)

```
static public String borrar (String x, String y) {
      String salida = "";
      int ini = 0, idx = 0;
      while (true) {
            idx = y.indexOf(x, ini);
            if (idx < 0) break;
            salida = salida + y.substring(ini, idx);
            ini = idx + x.lenght();
      }
      salida = salida + y.substring(ini);
      return salida;
}
```

b)

```
Console c = new Console();
c.print("Nombre del archivo de entrada?");
String inFile = c.readLine();
c.println("El resultado quedará en " + inFile + ".out");
BufferedReader in = new BufferedReader(new FileReader(inFile));
PrintWriter out = new PrintWriter(new FileWriter(inFile + ".out"));
while(true) {
      String line = in.readLine();
      if (line == null) break;
      line = borrar(":-)", line);
      line = borrar(":-(", line);
      out.println(line);
}
in.close();
out.close();
```

12. La Clase Secreto (Pregunta 3 Control 1, 2001)

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
`Secreto(int x,int y)`	Constructor que calcula y guarda un Nº secreto (entero entre x e y generado al azar)	`Secreto s = new Secreto(10,20)`
`int comparar(int x)`	-1 si Nº secreto < x 0 si Nº secreto = x 1 si Nº secreto > x	`s.comparar(15)`
`int menorMasCerca()`	Entrega el Nº menor más cercano que ha sido comparado con el Nº secreto	`s.menorMasCerca()`
`int mayorMasCerca()`	Entrega el Nº mayor más cercano que ha sido comparado con el Nº secreto	`s.mayorMasCerca()`

a) escribir un programa que utilice la clase anterior para implementar un juego en que el usuario tenga que adivinar un Nº 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 mas 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) { . . . }
}
```


Solución

a)

```
Console c = new Console();
Secreto num = new Secreto(1, 100);
c.println("Adivine un número entre 1 y 100");
int jugado = 0;
while (num.comparar(jugado) != 0) {
      c.print("X ? ") +
      jugado = c.readInt();
      if (jugado == 0) { // AYUDA
            c.println(num.menorMasCerca() + " < X < " +
                  num.mayorMasCerca());
      }
      else if (num.comparar(jugado) > 0) {
            c.println("X > "·+ jugado);
      }
      else if (num.comparar(jugado) < 0) {
            c.println("X < "·+ jugado);
      }
}
c.println("X = "·+ jugado + "   FELICITACIONES!");
```

b)

```
public int comparar(int x) {
      if (this.secreto < x) return -1;
      if (this.secreto > x) return -1;
      return 0;
}
```

13. División con Decimales (Pregunta 1 Control 1, 2002)

Al dividir dos números enteros se puede obtener un resultado con una cantidad determinada de decimales. Al respecto, escriba un programa que siga el diálogo indicado en el siguiente ejemplo:

```
Ingrese el dividendo : 22
Ingrese el divisor : 7
Ingrese la cantidad de decimales del resultado : 20
Resultado = 3.14285714285714285714
```

Nota. Los dígitos de la parte decimal se obtienen sucesivamente del resultado de la división entera entre el resto o residuo anterior (multiplicado por 10) y el divisor. Para el ejemplo:

dividendo	22	10	30	20	60	40	50	10	30	20	60	40	50	10	30	20	60	40	50	10	30
cuociente	3	1	4	2	8	5	7	1	4	2	8	5	7	1	4	2	8	5	7	1	4
resto	1	3	2	6	4	5	1	3	2	6	4	5	1	3	2	6	4	5	1	3	2

Solución

```
Console c = new Console();
c.print("Ingrese el dividendo?");
int dividendo = c.readInt();
c.print("Ingrese el divisor?");
int divisor = c.readInt();
c.print("Ingrese la cantidad de decimales del resultado?");
int decimales = c.readInt();
c.print("Resultado = " + (dividendo / divisor) + ".");
int resto = (dividendo % divisor) * 10;
for (int i=0; i<decimales; i++) {
      c.print((resto / divisor));
      resto = (resto % divisor) * 10;
}
```


14. Eliminación Delimitada (Pregunta 2 Control 1, 2002)

a) Escribir una función que elimine de un string los caracteres comprendidos entre dos delimitadores. Por ejemplo, eliminar("a/*bc*/d","/*","*/") entrega "ad". 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//c","//","") entrega "ab".

b) Utlice el método anterior 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 </html> en sus primeras columnas.

Solución

a)

```
public String eliminar(String txt, String del1, String del2) {
      String s = txt;

      while (true) {
            // Buscamos el primer delimitador
            int idx1 = s.indexOf(del1);
            if (idx1 < 0) break;
            s = s.substring(0, idx1);

            // Buscamos el segundo delimitador
            if (del2 == "") break;
            int idx2 = s.indexOf(del2);
            if (idx2 < 0) break;
            s = s.substring(idx2 + del2.length() + 1);
      }

      return s;
}
```

b)

```
Console c = new Console();
while(true) {
      String linea = c.readLine();
      if (linea.equals("</html>")) break;
      linea = eliminar(txt, "%", "");
      linea = eliminar(txt, "<", ">");
      c.println(linea);
}
```

15. Calendario (Pregunta 3 Control 1, 2002)

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 ≤ x ≤ 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 dia 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 los métodos mañana y nombre, suponiendo la siguiente declaración de la clase:

```
class Dia{
   private int d; // Representacion
   ...
}
```

Solución

a)

```
Console c = new Console();
c.print("Ingrese cantidad de dias del mes (1-31) : ");
int dias = c.readInt();
c.print("Primer dia del mes (1-7) : ");
int primer = c.readIne();
Dia d = new Dia(primer);
Dia DOM = new Dia(7);
for (int i=0; i<dias; i++) {
      if (d.comparar(DOM) == 0) {
            c.print(d.nombre.substring(0, 2).toUpperCase());
      }
      else {
            c.print(d.nombre.substring(0, 2));
      }
      c.println(" ");
      d = d.mañana();
}
```



```
c.println();
for (int i=0; i<dias; i++) {
      if (i < 10) {
            c.print(" ");
      }
      c.print(i + " ");
}
c.println();
```

b)

```
public Dia mañana() {
      Dia m = new Dia(this.d+1);
      return m;
}

public String nombre() {
      switch (this.d) {
            case 1:
                  return "lunes";
            case 2:
                  return "martes";
            case 3:
                  return "miércoles";
            case 4:
                  return "jueves";
            case 5:
                  return "viernes";
            case 6:
                  return "sábado";
            case 7:
                  return "domingo";
      }
      return null;
}
```

16. Cartas Modelo

Honorato Mayorga tiene varias pololas, y quiere automatizar la correspondencia que mantiene con ellas. En particular, tiene una carta escrita en el archivo "carta.txt" y desea adaptarla a las características de sus amigas. En la carta ha dejado con @ el nombre de pila y con # el color de ojos de cada amiguita.

Este archivo es como sigue:

```
Querida @:

Dejame decirte que me gustas tanto como antes, o
quizas mas que antes. El color # de tus ojos me inspira
mucho, de hecho, # es mi color favorito. Asi, @, por favor
responde mis llamadas. Mandame tu respuesta en un sobre #.

Por siempre tuyo
Honorato.
```

Pese a que Ud. ha desaconsejado a Don Honorato, el le ha pedido el favor especial de escribir un programa que solicite el nombre y el color de ojos de una dama, tome la carta del archivo "carta.txt" y escriba la carta resultante en un archivo de nombre "salida.txt", reemplazando el nombre y color de ojos de la dama donde corresponda. Nota: una linea de la carta puede requerir mas de un reemplazo.

Solución

```
class CartaModelo {
	static public void main (String[] args) {
		Console c = new Console();

		c.println ("Ingrese el nombre de la dama?");
		String nombre = c.readLine();
		c.println ("Ingrese su color de ojos?");
		String color = c.readLine();

		BufferedReader carta = new BufferedReader (
			new FileReader ("carta.txt"));
		PrintWriter salida = new PrintWriter (
			new FileWriter ("salida.txr"));
		String linea;
		while ( (linea = carta.readLine()) != null) {
			// Se reemplazan los @
			int anterior = 0;
			int indice = linea.indexOf("@");
			while (indica >= 0) {
				linea = linea.substring(anterior, indice)+
					nombre +
					linea.substring(indice + 1);
				anterior = indice + 1;
				indice = linea.indexOf("@");
			}

			// Se reemplazan los #
			anterior = 0;
			indice = linea.indexOf("#");
```



```
			while (indica >= 0) {
				linea = linea.substring(anterior, indice)+
					color +
					linea.substring(indice + 1);
				anterior = indice + 1;
				indice = linea.indexOf("@");
			}

			// Se escribe la linea en la salida
			salida.println(linea);
		}

		// Se cierran los archivos
		salida.close();
		carta.close();
	}
}
```

17. Palíndrome

Escribir un programa que lea una palabra y escriba si es o no capicúa (palíndrome), es decir, si se lee (o no) igual en ambos sentidos. Por ejemplo: "radar", "oso" y "alla" se escriben igual si se ponen al derecho o al revés.

Solución CON string auxiliar

```
class capicua {
        public static void main (String args[]) {
                Console c=new Console();

                c.print ("Palabra?");
                String palabra = c.readString();

                String aux = "";

                for (int i=palabra.length()-1; i >= 0; --i)
                        aux = aux + palabra.charAt(i);

                if (palabra.compareTo(aux) == 0)
                        c.println ("es capicua");
                else
                        c.println ("no es capicua");
        }
}
```

Solución SIN string auxiliar

```
public class capicua {
        public static void main (String args[]) {
                Console c=new Console();

                c.print ("Palabra?");
                String palabra = c.readString();

                int L = palabra.length();
                int i = 0;

                while (i < L/2 &&
                       palabra.charAt(i) == palabra.charAt(L-1-i))
                        i++;

                if ( i >= L/2 )
                        c.println ("es capicua");
                else
                        c.println ("no es capicua");
        }
}
```


18. Fracciones

(a) Escribir la clase Fraccion que contenga los siguientes métodos:

Ejemplo	Significado
Fraccion(No,No)	Constructor que recibe valores enteros para el número y el denominador
a.sumar(b)	sumar Fraccion b a Fraccion a
a.multiplicar(b)	Multiplicar Fraccion a por Fraccion b
a.comparar(b)	0 si a=b, No<0 si a0 si a>b
a.copiar(b)	copiar Fraccion b en Fraccion a
a.simplificar()	simplificar Fraccion a
a.toString()	entregar String con Fraccion a expresada en la forma No/No

(b) Escribir un programa, que utilice la clase Fraccion para leer una cantidad indeterminada de fracciones y escribir su promedio y la fracción de mayor valor, ambos como fracciones simplificadas. El programa debe responder al diálogo indicado en el siguiente ejemplo:

```
Promedio y Mayor de lista de fracciones
Fin de datos se indica con denominador cero
Numerador ?
Denominador ?
Suma = No/No
...
Numerador ?
Denominador ? 0
Promedio = No/No
Mayor = No/No
```

Solución

```
// Parte (a)
public class Fraccion {
       // representacion
       public int a,b;   // numerador y denominador

       // operaciones publicas
       public String toString() {
              return this.a + "/" + this.b;
       }
       public void sumar(Fraccion x) {
              this.a = this.a * x.b + b * x.a;
              this.b = this.b * x.b;
       }
       public void multiplicar(Fraccion x) {
              this.a = this.a * x.a;
              this.b = this.b * x.b;
       }
       public int comparar(Fraccion x)  {
              return this.a * x.b - this.b * x.a;
       }
       public void simplificar() {
              int mcd = Fraccion.MCD(this.a, this.b);
              this.a = a / mcd;
              this.b = b / mcd;
       }
       static public int MCD(int x, int y) {
              if (x % y == 0)
```

```
                return y;
        else if (x < y)
                return MCD(y, x);
        else
                return MCD(y, x % y);
    }
    public void copiar(Fraccion x) {
        this.a = x.a;
        this.b = x.b;
    }
    //constructor
    public Fraccion(int x, int y) {
        this.a = x;
        this.b = y;
    }
}
```

```
// Parte (b)
class Fracciones {
    static public void main (String args[]) throws IOException {
        Console I = new Console ();
        l.println("Promedio y mayor de lista de fracciones");
        l.println("fin de datos es denominador cero");

        Fraccion suma = new Fraccion(0,1);
        int n = 0; //contador
        Fraccion mayor = new Fraccion(-Integer.MAX_VALUE,1);

        while(true) {
            System.out.print("Numerador ? ");
            int numerador = I.readLine();

            System.out.print("Denominador ? ");
            int denominador = I.readLine();

            if (denominador == 0) break;

            Fraccion f =
                new Fraccion(numerador,denominador);

            suma.sumar(f);
            l.println("Suma = " + suma.toString());
            ++n;

            if (f.comparar(mayor) >= 0)
                mayor.copiar(f);
        }
        if (n>0) {
            Fraccion promedio = new Fraccion(1,n);
            promedio.multiplicar(suma);
            promedio.simplificar();
            l.println("Promedio=" + promedio.toString());
            l.println("Mayor=" + mayor.toString());
        }
    }
}
```


19. Programación de la TV

La siguiente tabla define los métodos ofrecidos por la clase Tiempo que permite realizar operaciones con instantes de tiempo:

Ejemplo	significado	encabezamiento
a.suma(b)	a + b	Tiempo suma(Tiempo x)
a.resta(b)	a - b	Tiempo resta(Tiempo x)
a.comp(b)	0 si a=b, <0 si a<b, >0 si a>b	int comp(Tiempo x)
a.toString()	Entregar tiempo en la forma “HH:MM”	String toString()
a.minutos()	Entregar tiempo en minutos	int minutos()
new Tiempo()	Objeto con valor cero (0 hrs y 0 minutos)	Tiempo()
new Tiempo(h,m)	Objeto con valor de h horas y m minutos	Tiempo(int x, int y)
new Tiempo(“HH:MM”)	Objeto con valor HH hras y MM minutos	Tiempo(String x)
new Tiempo(b)	Objeto con mismo valor de objeto b	Tiempo(Tiempo x)

(a) Escribir un programa que use la clase Tiempo para entregar los siguientes resultados con respecto a la programación de un canal de TV:

- El nombre del programa más largo (y su duración en minutos)
- El nombre del programa más corto (y su duración en minutos)
- El promedio de duración de los programas (en minutos)

La programación del canal se lee a través de la entrada estándar de Java. Cada línea contiene la siguiente información de un programa:

- Hora de comienzo del programa (5 caracteres en la forma HH:MM, por ejemplo “21:00”)
- Nombre del programa (siguientes caracteres hasta el final de la linea, por ejemplo “Noticiario”)

Notas:

La programación se encuentra ordenada cronológicamente
La primera línea contiene el primer programa (por ejemplo: 07:00Noticas al despertar)
La última contiene la hora de cierre (por ejemplo 01:45Cierre de trasmisiones)

(b) Escribir la clase Tiempo

Solución

```
// Parte (a)
class TV{
static public void main(String[] args)throws IOException{
      //objetos para mantener promedio, mayor y menor
      Tiempo tMax =  new Tiempo(0,0);
Tiempo tMin =  new Tiempo(24,0);
Tiempo total = new Tiempo(0,0);
      String programaMax="", programaMin="";
      int n = 0;   //numero de programas
```

```
      //obtener primer programa y su hora de comienzo
Console in = new Console();
      String linea = in.readLine();
      Tiempo comienzoAnterior = new Tiempo(linea.substring(0,5));
      String programaAnterior = linea.substring(5);

      //leer lineas hasta "Ciere de trasmisiones"
      while( linea.equals("Cierre de trasmisiones") == false ){
            linea = in.readLine();

            //obtener programa y hora de comienzo
            Tiempo comienzo = new Tiempo(linea.substring(0,5));
            String programa = linea.substring(5);

            //ajustar hora de comienzo si pasa la medianoche
            if( comienzo.comp(comienzoAnterior) < 0 )
                  comienzo = comienzo.suma(new Tiempo(24,0));

            //calcular duracion del programa
            Tiempo t = comienzo.resta(comienzoAnterior);

            //mantener suma y cuenta de duraciones
            total = total.suma(t); ++n;

            //mantener programa de mayor duracion
            if(t.comp(tMax) > 0){
                  tMax=t; programaMax=programaAnterior;
            }
            //mantener programa de menor duracion
            if( t.comp(tMin) < 0 ){
                  tMin=t; programaMin=programaAnterior;
            }
            //actualizar programa anterior
            comienzoAnterior = comienzo;
            programaAnterior = programa;
      }
      //mostrar resultados
      in.println("Promedio="+ total.minutos()/n);
      in.println("Mayor="+ tMax.minutos() +
 " programa=" + programaMax);
      in.println("Menor="+ tMin.minutos() +
 " programa=" + programaMin);
}
}
```

// parte (b)

```
public class Tiempo {
//representacion
public int h,m;     //horas y minutos

//operaciones
public int comp(Tiempo x) {
      return (60*this.h+this.m) - (60*x.h+x.m);
}
public String toString() {
      return this.h + ":" + this.m;
}
public Tiempo suma(Tiempo x)    {
      return new Tiempo(this.h+x.h, this.m+x.m);
}
public Tiempo resta(Tiempo x) {
```



```
      return new Tiempo( this.minutos() - x.minutos() );
}
public int minutos() {
      return  60*this.h + this.m;
}

//constructores (inicializadores) de objetos de clase Tiempo
public Tiempo(int x, int y) {
      this.h=x+y/60;
      this.m=y%60;
}
public Tiempo(int x) {
      this.h=x/60;
      this.m=x%60;
}
public Tiempo()    {
      this.h=0;
      this.m=0;
}
public Tiempo(Tiempo x) {
      this.h=x.h;
      this.m=x.m;
}
public Tiempo(String x)   {
      int i = x.indexOf(":");
      h = Integer.parseInt(x.substring(0,i));
      m = Integer.parseInt(x.substring(i+1));
}
}
```

20. Fechas

La clase Fecha contiene los métodos indicados en la siguiente tabla:

Ejemplo (a y b son Fechas)	Significado	Encabezamiento
a.resta(b)	a – b en años	int resta(Fecha x)
a.comp(b)	0 si a=b, Nº<0 si a<b, Nº>0 si a>b	int comp(Fecha x)
new Fecha("dd/mm/aaaa")	Objeto con fecha dd/mm/aaaa	Fecha(String x)

(a) Escriba un programa que use la clase para calcular los años transcurridos entre dos fechas, de acuerdo al diálogo indicado en el siguiente ejemplo:

```
Calcular diferencia en años entre dos fechas
Fecha 1 (dd/mm/aaaa) ? 22/01/2001
Fecha 2 (dd/mm/aaaa) ? 24/08/1989
Años transcurridos = 11
```

Nota: Las fechas pueden estar en cualquier orden.

(b) Escribir el método comp, suponiendo la siguiente declaración de la clase Fecha:

```
public class Fecha {
      public int d, m, a; //representación: dia, mes y año
      . . .
}
```

(c) Propuesto. Escribir los métodos resta y constructor

Solución

```
      //contadores de preguntas y puntos
      int n=0, puntos=0;

      //leer lineas hasta fin de archivo
      Consoles in = new Console();
      BufferedReader A = new BufferedReader(
            new FileReader("preguntas.txt"));

      while(true)
      {
            String linea = A.readLine();
            if( linea == null ) break;

            //mostrar pregunta
            c.println("Pregunta "
                  + (++n)+":"+linea.susbtring(1));

            //obtener respuesta
            c.print("V o F ? ");
            String s = in.readLine();

            //contar y mostrar puntos
            if( s.equals(linea.substring(0,1)) )
                  ++puntos;
            c.println("Puntos = " + puntos);
      }
```



```
-----------------------------------------------------------------------
      //obtener fecha 1
      c.print("Fecha 1 (dd/mm/aaaa) ? ");
      Fecha1 f1 = new Fecha(in.readLine());

      //obtener fecha 2: 0.75
      c.print("Fecha 2 (dd/mm/aaaa) ? ");
      Fecha1 f1 = new Fecha(in.readLine());

      //calcular y mostrar diferencia
      if( f1.comp(f2) < 0 )
            c.println( "An~os=" + f1.resta(f2) );
      else
            c.println( "An~os=" + f2.resta(f1) );
-----------------------------------------------------------------------
public int comp(Fecha x)
{
      //comparar años
      if( a < x.a ) return -1;
      if( a > x.a ) return 1;
      //comparar meses
      if( m < x.m ) return -1;
      if( m > x.m ) return 1;
      //comparar dias
      return d - x.d;
}
}
```

21. Servicio de Inteligencia

El servicio de inteligencia de Putre, le ha solicitado a usted que programe un traductor del codigo Morse, para lo cual le proporciona la clase Morse, que le permite traducir una letra del ya mensionado codigo al alfabeto que usamos.

Encabezado	Descripción	Ejemplo
public Morse()	Constructor	Morse m = new Morse();
public char toAlfabeto(String clave)	Traduce de clave Morse a alfabeto	String b = m.toAlfabeto(".-")

(a)Usted debe construir la clase Traductor, con su respectivo constructor, y los métodos morseToAlfa, que recibe un string en clave morse, con cada letra separada por un "/", y retorna un string con la traduccion. Y el método traducirArchivo, que recibe el nombre del archivo a traducir, y el nombre del archivo donde debe guardar la traducción.

i.e.

Encabezado	Descripción	Ejemplo de uso
Public Traductor();	Constructor	Traductor t = new Traductor();
Public String morseToAlfa(String mensaje)	Traduce de Morse al alfabeto	String message = t.morseToAlfa(elmensaje);
public void traducirArchivo(String mensaje, String traducido)	Traduce un archivo Morse y guarda en otro archivo	t.traducirArchivo("mensaje.txt", "mensajetrad.txt"

Solución

```
import java.io.*;

public class  Traductor {

	private Morse m;

	public Traductor (){
		m = new Morse();
	}

	public String morseToAlfa(String mensaje) {
		String traduccion = "";
		int i = 0;
		int j = mensaje.indexOf("/");
		while (j != -1) {
			traduccion += m.toAlfabeto(
				mensaje.substring(i,j));
			i = j+1;
			j = mensaje.indexOf("/", i);
		}
		traduccion += m.toAlfabeto(mensaje.substring(i));
		return traduccion;
	}
```



```
	public void traducirArchivo(String mensaje,
			String traducido)throws IOException {
		BufferedReader L = new BufferedReader(
			new FileReader(mensaje));
		PrintWriter E = new PrintWriter(
			new FileWriter(traducido));
		String linea;
		while ((linea = L.readLine()) != null) {
			E.println(morseToAlfa(linea));
		}
		L.close();
		E.close();
	}

}
```

22. Notas de Controles

En CC10A, las notas de los controles se guardan en archivos CONTROLES_xx.TXT en donde xx indica el número de la sección de computación. Por ejemplo, las notas de la sección 06 estaría en el archivo CONTROLES_06.TXT.

El formato de estos archivos es bastante peculiar y es el siguiente:

<username>:<nombre completo>:<codigo alumno>:<c1>:<c2>:<c3>:<c4>:<c5>

Cada campo está separado por ":". Por ejemplo, el archivo puede ser:

```
aacuna:Angel Acuna Avila:254:3.7:2:7:5.3:6.4
aalvarez: Avelina Alvarez Barraza:123:7:6:5.3:1:1
...
```

Pero este formato no siempre posee todas las notas de los controles. Es decir, a medida que se van realizando los controles, los campos en cada línea del archivo van aumentando. Es así como a principio de año solo se observan 3 campos:

```
joconcha:Jorge Concha Concha:12
```

y durante el año este va creciendo. Para el control 1:

```
joconcha:Jorge Concha Concha:12:7
```

el control 2:

```
joconcha:Jorge Concha Concha:12:7:4.3
```

etc, hasta el control 5:

```
joconcha:Jorge Concha Concha:12:7:4.3:1:2.7:6.5
```

El procesamiento de estos archivos requiere un gran trabajo por parte de los profesores y es por eso que necesitan de algo que les automatice PARA CUALQUIER SECCION este trabajo.

Un Arquitecto de Software que conoce el concepto de Orientación al Objeto (pero no cacha cómo programar) definió que la forma de procesar esto era creando un "parser" que trabajara con cada línea por separado (sin preocuparse de donde provenía) y que utilizando las funcionalidades de este parser el usuario pudiera obtener el i-ésimo parámetro o saber cuántas notas trae el archivo. Con esta premisa definió una clase Parser con las siguientes características:

ELEMENTO	DESCRIPCION
String linea	Variable de Instancia que representa la línea con el cual el parser está trabajando.
Parser(String)	Constructor que permite entregarle al parser cuál será la línea con la cual debe trabajar.
String saca(int)	Método que recibe un entero que indica el número del parámetro que debe retornar (partiendo de 1). Si el parámetro no está, retorna *null*.
int cuenta()	Método que permite saber cuántos campos tiene la línea.

a)Escriba la clase Parser con todos sus métodos.

b)Escriba un programa principal que responda al siguiente diálogo:

```
Ingrese sección? 06
PROMEDIOS DE LA SECCION 06
Angel Acuna Avila - 3.7
Avelina Alvarez Barraza - 5.7
...
Total: 97 alumnos
```

Nota: No es necesario hacer la parte (a) para hacer la parte (b)

Solución parte (a)

Para esta parte, lo que es necesario es crear un archivo llamado Parser.java que contiene esta clase:

```
public class Parser {
      // Se declara la variable de instancia de la clase.
      public String linea;

      // Se escribe ahora el constructor
      public Parser (String linea) {
            // Lo único que tenemos que hacer es tomar lo que nos
            // entregan como parte del constructor y llevarlo a la
            // variable de instancia con un ":" al final.
            this.linea = linea + ":";
      }

      // Se escribe el método que saca los elementos
      public String saca(int n) {
            // En este método tenemos que sacar el n-ésimo elemento
            // de la línea this.linea.

            // Primero verificamos que no sea nula la linea
            if (this.linea == null)
                  return null; // idem que si no está

            // Estado inicial del elemento
            String elto = null;
```

```
            // Definimos cuales serán los márgenes del primer
            // elemento
            int inicio = 0;
            int fin = this.linea.indexOf(":");

            // Buscamos hasta completar n veces
            for (int i=1; i<n && fin>=0; i++) {
                  inicio = fin + 1;
                  fin = this.linea.indexOf(":", inicio);
            }

            // Ahora sacamos el elemento correcto si es que está en
            // la linea
            if (fin > 0)
                  elto = this.linea.substring(inicio, fin);

            // Retornamos el elemento encontrado
            return elto;
      }

      // Se escribe el método que cuenta los campos de la linea
      public int cuenta() {
            // Como le pusimos un ":" al final, sabemos que la
            // cantidad de campos es IGUAL a la cantidad de ":"
            // que hayan en la línea, por lo que podemos contar
            // los ":".

            // Primero verificamos que no sea nula la linea
            if (this.linea == null)
                  return 0; // no hay elementos

            // Estado inicial del contador
            int n = 0;

            // Definimos cuales será la posición del primer ":"
            int pos = this.linea.indexOf(":");

            // Buscamos mientras hayan ":"
            while (pos >= 0) {
                  n++; // Se incrementa el contador en 1
                  pos = this.linea.indexOf(":", pos + 1);
            }

            // Retornamos la cantidad de elementos encontrada
            return n;
      }
}
```

Solución parte (b)

Para la parte b es necesario crear un nuevo archivo .java para el programa principal. En estricto rigor, en el control no es necesario hacerlo, pero para que funcione, deben hacerlo así.

```
import java.io.*;

public class ProgramaPrincipal {
      static public void main(String[] args) throws Exception {
            // Declaramos nuestra consola de siempre
            Console c = new Console();

            // Pedimos la sección
```



```
            // solo así puedo aguantar 06
            c.print("Ingrese la sección ? ");
            String seccion = c.readLine();

            // Encabezado
            c.println("PROMEDIOS DE LA SECCION 06");

            // Abrimos el archivo correcto
            BufferedReader arc = new BufferedReader(
                new FileReader("CONTROLES_" + seccion + ".TXT"));

            // Leemos línea a línea y la procesamos usando el
            // parser
            int nAlumnos = 0;
            while (true) {
                  String linea = arc.readLine();

                  // Si llegamos al fin de archivo, terminamos
                  if (linea == null) break;

                  // Creamos el parser para la línea
                  Parser lineaP = new Parser(linea);

                  // Sacamos el nombre del alumno y la cantidad de
                  // notas que hay en la línea
                  String nombre = lineaP.saca(2);
                  int cant = lineaP.cuenta() - 3;
                  double promedio = 0;

                  // Si existen notas, sacamos las notas y las
                  // promediamos
                  if (cant > 0) {
                        for (int i=1; i<=cant; i++) {
                              // Se saca la nota transformándola
                              // a double
                              promedio += new Double(
                                lineaP.saca(i+3)).doubleValue();
                        }

                        // Se promedia la suma que calculamos
                        promedio = promedio / cant;

                        // Se imprimen los datos del alumno
                        c.println(nombre + " - " + promedio);
                  }
                  else {
                        // Se imprimen un mensaje indicando que el
                        // alumno no tiene datos
                        c.println(nombre + " - SIN NOTAS");
                  }

                  // Se incrementa el número de alumnos
                  nAlumnos++;
            }

            // Se imprime el total de alumnos
            c.println("Total: " + nAlumnos + " alumnos");
      }
}
```