PROGRAMACIÓN I 6-febrero-2004

1. *(2 puntos)* Dado el siguiente fragmento de programa:

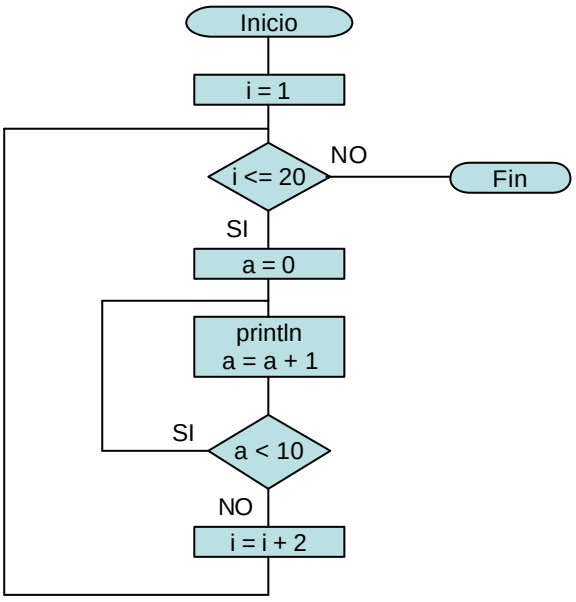
```
for (int i = 1; i <= 20; i = i +2)
{
    a = 0;
    do
    {
            System.out.println (i + " " + a);
            a = a + 1;
    } while (a < 10);
}
```

a. ¿Cuántas veces se ejecuta la instrucción System.out.println?
b. ¿Cuál es el diagrama de flujo correspondiente a este fragmento?

Solución del apartado a):

El bucle exterior se ejecuta 10 veces y por cada iteración del bucle exterior el bucle interior se ejecuta 10 veces. Por tanto, dicha instrucción se ejecuta 100 veces.

Solución del apartado b):



2. *(8 puntos)* El objetivo de la criptografía es someter un texto original a una transformación mediante una clave de cifrado, obteniéndose de esta forma un texto cifrado. A esta transformación se le denomina encriptación o cifrado. A lo largo de la historia, y ya desde los tiempos de Julio César, se han ido desarrollando diferentes y cada vez más complejos métodos de cifrado. Estos métodos se pueden clasificar en dos grandes grupos. En los métodos de sustitución un carácter o grupo de caracteres se cambia por otro u otros diferentes. En los métodos de transposición o de permutación un grupo de caracteres sufre cambios de posición con respecto a su posición original.

El programa consiste en realizar un encriptador basado en el método de permutación columnar. Se trata de dividir el texto original en grupos de caracteres del mismo tamaño que la clave de cifrado. Sobre cada uno de estos grupos se aplica la permutación de los caracteres en la forma que la clave de cifrado lo indique.

Por ejemplo, si se dispone de la clave de cifrado de 8 caracteres numéricos

75310246

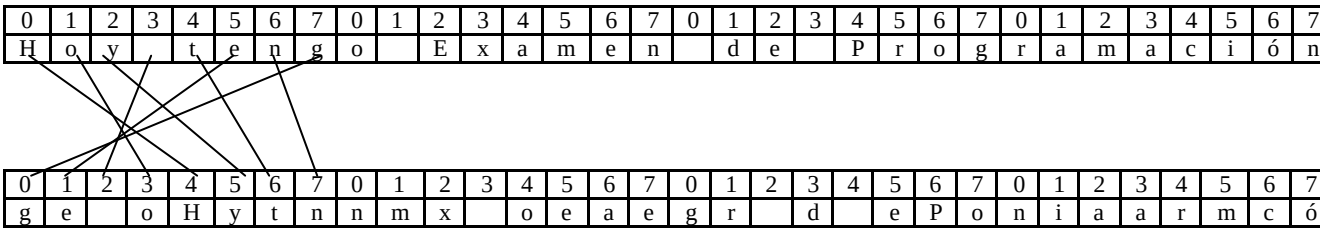
y el texto original es

Hoy tengo examen de Programación

entonces el texto cifrado resultante es

ge oHytnnmx oeaegr d ePoniaarmcó

Es decir:



Como se puede apreciar el método de encriptación consiste en agrupar los caracteres del texto original de 8 en 8, y sobre cada uno de los grupos aplicar la transposición reflejada en la clave de cifrado. En otras palabras, el carácter que ocupa la posición 0 pasa a la posición 4, el que ocupa la posición 1 pasa a la posición 3, etc.

a) Se pide realizar un programa codificado en lenguaje Java y utilizando el paquete javabook, teniendo en cuenta las siguientes condiciones:

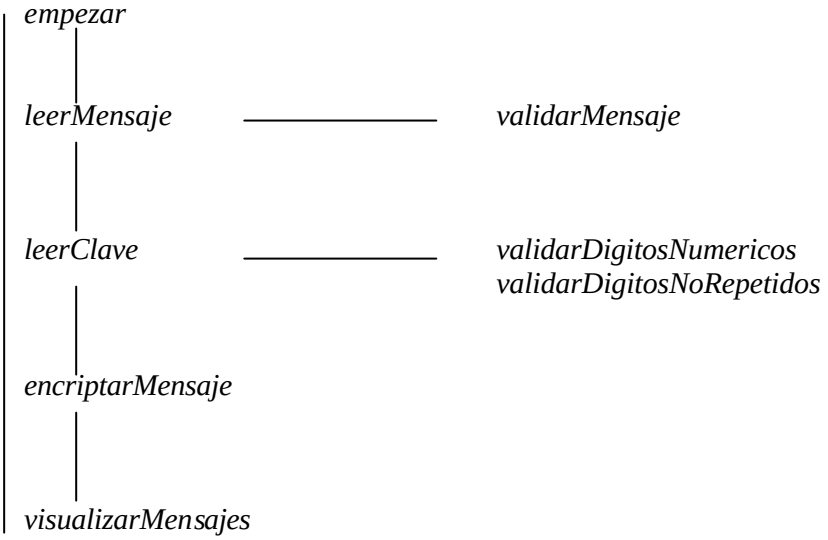
- El texto original tecleado es una cadena de caracteres y debe ser de una longitud múltiplo de 8.
- La clave de cifrado introducida es una secuencia de 8 caracteres numéricos y debe contener dígitos que pertenezcan al intervalo [0,7] y que no se repitan entre ellos (cada dígito es un número no repetido del 0 al 7).
- En caso de error en el texto original, se debe visualizar el siguiente mensaje:

  El mensaje debe ser de longitud múltiplo de 8.
- En caso de error en la clave de cifrado, se debe/n visualizar el/los siguiente/s mensaje/s:

  Los dígitos de la clave deben pertenecer a [0,7]. Repetir.

  Los dígitos de la clave están repetidos. Repetir.
- Se deben seguir introduciendo el texto original y la clave de cifrado, mientras sean incorrectos, considerando los errores anteriores.

- Cuando sean correctos, se deben visualizar por pantalla el mensaje original, la clave de encriptación y el mensaje cifrado.
- El programa finaliza cuando el usuario responde con el botón NO a la siguiente pregunta:
  ¿Quieres encriptar mensajes según el método de permutación columnar?
- Hay que implementar dos clases denominadas *PermutacionColMain y PermutacionCol.* El diseño de la clase *PermutacionCol* debe incluir como atributos los objetos de la clase javabook necesarios, el texto original, la clave de encriptación y el texto cifrado.
- Completar el prototipo de los métodos, que obligatoriamente deben cumplir el siguiente esquema:



b) Se pide modificar todo el programa anterior, si el diseño de la clase *PermutacionCol* sólo incorpora como atributos los objetos de la clase javabook necesarios, el texto original y la clave de encriptación.

Solución del apartado a):

```
import javabook.*;

class PermutacionCol
{
    //atributos
    private MainWindow ventanaPrincipal;
    private InputBox cuadroEntrada;
    private OutputBox cuadroSalida;
    private MessageBox cuadroMensaje;
    private ResponseBox cuadroRespuesta;
    private String mensajeEntrada;
    private String claveCifrado;
    private String mensajeSalida;

    //constructor
    public PermutacionCol()
    {
      ventanaPrincipal = new MainWindow();
            cuadroEntrada = new InputBox (ventanaPrincipal);
            cuadroSalida = new OutputBox (ventanaPrincipal);
```

```
            cuadroMensaje = new MessageBox (ventanaPrincipal);
            cuadroRespuesta = new ResponseBox (ventanaPrincipal);

            ventanaPrincipal.setVisible(true);
            cuadroSalida.show();
    }

    //metodo empezar
    public void empezar()
    {
            int respuesta;
            respuesta = cuadroRespuesta.prompt("¿Quieres encriptar
mensajes según el método de permutación columnar?");
            while (respuesta == cuadroRespuesta.YES)
            {

            leerMensaje();
            leerClave();

            encriptarMensaje();

            mostrarMensajes();

            respuesta = cuadroRespuesta.prompt("¿Quieres encriptar
mensajes según el método de permutación columnar?");
            }
    }

    //método leerMensaje
    private void leerMensaje()
    {
      do
            {
                  mensajeEntrada = cuadroEntrada.getString("Introduce
el mensaje:");
                  if (validarMensaje() == false)
                        cuadroMensaje.show("El mensaje debe ser de
longitud múltiplo de 8.");
            } while (validarMensaje() == false);
      }

      //método validarMensaje
      private boolean validarMensaje()
      {
            boolean esCorrecto = true;
            if (mensajeEntrada.length() % 8 != 0)
                  esCorrecto = false;
            return esCorrecto;
    }

    //método leerClave
    private void leerClave()
    {
            boolean esNumerica = true;
            boolean esNoRepetida = true;

            do
            {
                  claveCifrado = cuadroEntrada.getString ("Introduce la
clave: ");
```

```
                esNumerica = validarDigitosNumericos();
                esNoRepetida = validarDigitosNoRepetidos();

                if (esNumerica == false)
                cuadroMensaje.show("Los dígitos de la clave deben
pertenecer a [0,7]. Repetir");

                if (esNoRepetida == false)
                cuadroMensaje.show("Los dígitos de la clave están
repetidos. Repetir");

          } while (!esNumerica || !esNoRepetida);
    }

    //método validarDigitosNumericos
      private boolean validarDigitosNumericos()
      {
            int i = 0;
            boolean claveNumerica = true;

            while (i < 8 && claveNumerica)
            {
                  if (claveCifrado.charAt(i) < '0' ||
claveCifrado.charAt(i) > '7')
                        claveNumerica = false;
                  else
                        i++;
            }
            return claveNumerica;
      }

    //método validarDigitosNoRepetidos
    private boolean validarDigitosNoRepetidos()
    {
            char carActual;
            int i = 0, j = 0;
            boolean claveNoRepetida = true;
            do
            {
                  i++;
                  carActual = claveCifrado.charAt(i);
                  j = 0;

                  while (j < i && claveNoRepetida)
                  {
                        if (carActual == claveCifrado.charAt(j))
                              claveNoRepetida = false;
                        else
                              j++;
                  }

            } while (i < 7 && claveNoRepetida);

            return claveNoRepetida;
      }

    //método encriptarMensaje
    private void encriptarMensaje()
    {
            StringBuffer objetoTemp = new StringBuffer();
            int longitud = mensajeEntrada.length();
```

```
            String letraClave = null;
            int indice;
            char caracterEncriptado;

            for (int i = 0; i < longitud; i = i + 8)
            {
                  for (int j = 0; j < 8; j++)
                  {
                        letraClave = claveCifrado.substring(j,j+1);
                        indice = Integer.parseInt(letraClave);
                        indice = indice + i;
                        caracterEncriptado =
mensajeEntrada.charAt(indice);
                        objetoTemp.append(caracterEncriptado);
                  }
            }
            mensajeSalida = objetoTemp.toString();
    }

    //método mostrarMensajes
    private void mostrarMensajes()
    {
      cuadroSalida.printLine ("Mensaje de entrada: " +
mensajeEntrada);
      cuadroSalida.printLine ("CLAVE: " + claveCifrado);
      cuadroSalida.printLine ("Mensaje de salida:  " + mensajeSalida);
    }
}

class PermutacionColMain
{
    public static void main (String[] args)
    {
        PermutacionCol permu = new PermutacionCol( );
        permu.empezar();
    }
}
```

Solución del apartado b) mostrando sólo los métodos que incluyen los cambios necesarios con respecto al apartado anterior:

```
class PermutacionCol2
{
    //atributos
    private MainWindow ventanaPrincipal;
    private InputBox cuadroEntrada;
    private OutputBox cuadroSalida;
    private MessageBox cuadroMensaje;
    private ResponseBox cuadroRespuesta;
    private String mensajeEntrada;
    private String claveCifrado;
    //private String mensajeSalida;

    //metodo empezar
    public void empezar()
    {
            String mensajeSalida = null;
            int respuesta;
            respuesta = cuadroRespuesta.prompt("¿Quieres encriptar
mensajes según el método de permutación columnar?");
```

```
            while (respuesta == cuadroRespuesta.YES)
            {

            leerMensaje();
            leerClave();

            mensajeSalida = encriptarMensaje();

            mostrarMensajes(mensajeSalida);

            respuesta = cuadroRespuesta.prompt("¿Quieres encriptar
mensajes según el método de permutación columnar?");
            }
    }

    //método encriptarMensaje
    private String encriptarMensaje()
    {
            StringBuffer objetoTemp = new StringBuffer();
            int longitud = mensajeEntrada.length();
            String letraClave = null;
            int indice;
            char caracterEncriptado;

            for (int i = 0; i < longitud; i = i + 8)
            {
                  for (int j = 0; j < 8; j++)
                  {
                        letraClave = claveCifrado.substring(j,j+1);
                        indice = Integer.parseInt(letraClave);
                        indice = indice + i;
                        caracterEncriptado =
mensajeEntrada.charAt(indice);
                        objetoTemp.append(caracterEncriptado);
                  }
            }
            return objetoTemp.toString();
    }

    //método mostrarMensajes
    private void mostrarMensajes(String mensajeSalida)
    {
      cuadroSalida.printLine ("Mensaje de entrada: " +
mensajeEntrada);
      cuadroSalida.printLine ("CLAVE: " + claveCifrado);
      cuadroSalida.printLine ("Mensaje de salida:  " + mensajeSalida);
    }
}
```