

1. (1 punto) Dado el siguiente fragmento en Java:

```
class Pares
{
    public static void main (String [] argumentos)
    {
        String numero = argumentos[0];
        ...
    }
}
```

Completar el programa con las sentencias necesarias, para que a partir de un número entero positivo ($n > 0$) cualquiera, se visualicen todos los números pares posibles con la condición de que la suma de dichos números sea menor que el número n tecleado. Por ejemplo, si $n = 25$, se deben visualizar los pares:

```
2
4
6
8
```

Solución:

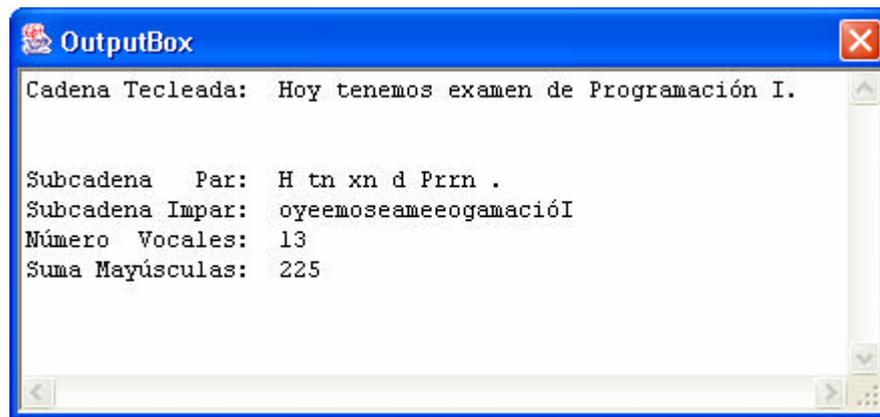
```
class Pares
{
    public static void main (String [] argumentos)
    {
        String numero = argumentos[0];
        int n = Integer.parseInt(numero);

        int par = 2;
        int suma = 2;

        while (suma < n)
        {
            System.out.println(par);
            par += 2;
            suma += par;
        }
    }
}
```

2. (4 puntos) Cualquier lenguaje de programación contiene un conjunto de funciones que se pueden aplicar a diferentes tipos de datos. Por ejemplo, en Java dada una cadena String se puede obtener la misma cadena String en mayúsculas aplicando el método `toUpperCase()`.

El objetivo del programa consiste en codificar un nuevo conjunto de operaciones para el TAD (Tipo Abstracto de Datos) denominado String, y en verificar posteriormente su aplicación. Estas operaciones vienen definidas por las necesidades del usuario y se explican más adelante. Una vez comprobado que la cadena introducida es correcta y aplicadas las operaciones, se debe visualizar la siguiente información:



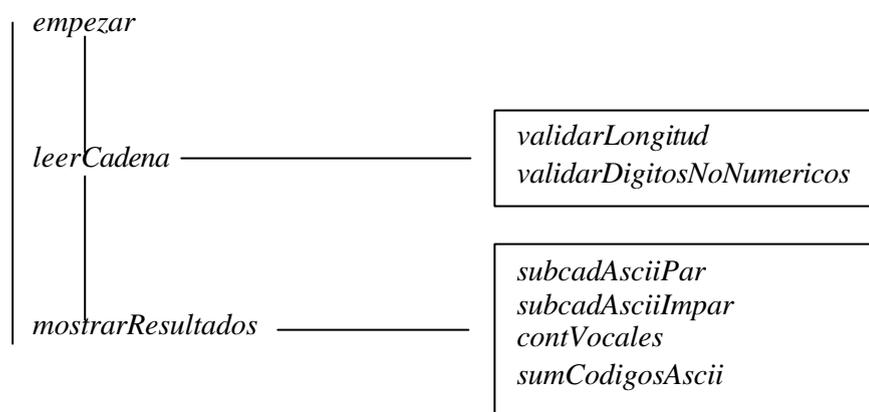
Se pide realizar un programa codificado en lenguaje Java, teniendo en cuenta las siguientes condiciones:

- La cadena tecleada es una cadena no numérica, que debe ser de longitud superior a 50 y que no debe contener caracteres que pertenezcan al intervalo [0,9].
- En caso de error en la cadena se debe/n visualizar el/los siguiente/s mensaje/s:
La longitud de la cadena debe ser como máximo igual a 50.
La cadena no puede tener caracteres numéricos.
Y seguir tecleando la cadena hasta que sea correcta.
- Las dos primeras operaciones a implementar se denominan `subcadAsciiPar()` y `subcadAsciiImpar()`. A partir de la cadena introducida se trata de obtener dos subcadenas diferentes, en función de que el código ASCII de cada carácter no numérico sea par o impar.
- En el siguiente ejemplo se indica cómo se obtienen dichas subcadenas:
Cadena = Hoy tenemos examen de Programación I.
Como la letra 'H' tiene por código ASCII el valor 72 forma parte de la subcadena par. Por el contrario, como la letra 'o' tiene por código ASCII el valor 111 forma parte de la subcadena impar. Y así con el resto de los caracteres de la cadena.
Subcadena Par = H...
Subcadena Impar = o...
- El tercer método a implementar se llama `contVocales()`. Tal como su nombre indica, se trata de contabilizar el número de vocales minúsculas y mayúsculas contenidas en la cadena tecleada.

- El cuarto método a implementar se llama `sumCodigosAscii()`. Tal como su nombre indica, se trata de sumar el valor del código ASCII de aquellos caracteres que en la cadena tecleada estén en mayúsculas.
- Cuando la cadena sea correcta, se deben visualizar por pantalla los resultados de aplicar los métodos descritos anteriormente a dicha cadena.
- El programa finaliza cuando el usuario responde con el botón NO a la siguiente pregunta:

¿Quieres repetir este programa?

- Hay que implementar dos clases denominadas *MiString* y *MiStringMain*. El diseño de la clase *MiString* debe incluir como atributos los objetos de la clase `javabook` necesarios y la cadena de caracteres tecleada.
- Completar el prototipo de los métodos, que obligatoriamente deben cumplir el siguiente esquema:



Solución:

```

package septiembre;

import javabook.*;

class MiString
{
    //atributos
    private MainWindow ventanaPrincipal;
    private InputBox cuadroEntrada;
    private OutputBox cuadroSalida;
    private MessageBox cuadroMensaje;
    private ResponseBox cuadroRespuesta;
    private String cadena;

    //constructor
    public MiString()
    {
        ventanaPrincipal = new MainWindow();
        cuadroEntrada = new InputBox (ventanaPrincipal);
        cuadroSalida = new OutputBox (ventanaPrincipal);
        cuadroMensaje = new MessageBox (ventanaPrincipal);
        cuadroRespuesta = new ResponseBox (ventanaPrincipal);
        cadena = new String();

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

```

//método empezar
public void empezar()
{
    int respuesta;
    respuesta = cuadroRespuesta.prompt("¿Quieres repetir este
programa?");

    while (respuesta == cuadroRespuesta.YES)
    {
        leerCadena();
        mostrarResultados();

        respuesta = cuadroRespuesta.prompt("¿Quieres repetir este
programa?");
    }
}

//método leerCadena
private void leerCadena()
{
    boolean esLongitudCorrecta = true;
    boolean esNoNumerica = true;

    do
    {
        cadena = cuadroEntrada.getString ("Introduce una cadena no
numérica: ");

        esLongitudCorrecta = validarLongitud();
        if (esLongitudCorrecta == false)
            cuadroMensaje.show("La longitud de la cadena debe ser
como máximo igual a 50. Repetir");

        esNoNumerica = validarDigitosNoNumericos();
        if (esNoNumerica == false)
            cuadroMensaje.show("La cadena no puede tener
caracteres numéricos. Repetir");

    } while (!esLongitudCorrecta || !esNoNumerica);
}

//método validarLongitud
private boolean validarLongitud()
{
    boolean esCorrecto = true;
    if (cadena.length() > 50)
        esCorrecto = false;
    return esCorrecto;
}

//método validarDigitosNoNumericos
private boolean validarDigitosNoNumericos()
{
    int i = 0;
    boolean cadenaNoNumerica = true;

    while (i < cadena.length() && cadenaNoNumerica)
    {
        if (cadena.charAt(i) >= '0' && cadena.charAt(i) <= '9')
            cadenaNoNumerica = false;
    }
}

```

```

        else
            i++;
    }
    return cadenaNoNumerica;
}

//método subcadAsciiPar
private String subcadAsciiPar()
{
    String subcadenaPar = new String();
    for (int i = 0; i < cadena.length(); i++)
    {
        char letra = cadena.charAt(i);
        int codigo = (int) letra;
        if (codigo % 2 == 0)
            subcadenaPar = subcadenaPar + letra;
    }
    return subcadenaPar;
}

//método subcadAsciiImpar
private String subcadAsciiImpar()
{
    String subcadenaImpar = new String();
    for (int i = 0; i < cadena.length(); i++)
    {
        char letra = cadena.charAt(i);
        int codigo = (int) letra;
        if (codigo % 2 != 0)
            subcadenaImpar = subcadenaImpar + letra;
    }
    return subcadenaImpar;
}

//método contVocales
private int contVocales()
{
    int total = 0;
    for (int i = 0; i < cadena.length(); i++)
    {
        char letra = cadena.charAt(i);
        if (letra == 'A' || letra == 'a' ||
            letra == 'E' || letra == 'e' ||
            letra == 'I' || letra == 'i' ||
            letra == 'O' || letra == 'o' ||
            letra == 'U' || letra == 'u')
            total = total + 1;
    }
    return total;
}

//método sumCodigosAscii
private int sumCodigosAscii()
{
    int total = 0;
    for (int i = 0; i < cadena.length(); i++)
    {
        char letra = cadena.charAt(i);
        if (letra >= 'A' && letra <= 'Z' || letra == 'Ñ')
        {
            int codigo = (int) letra;

```

```

        total = total + codigo;
    }
}
return total;
}

//método mostrarResultados
private void mostrarResultados()
{
    cuadroSalida.println ("Cadena Tecleada: " + cadena);
    cuadroSalida.skipLine(2);
    cuadroSalida.println ("Subcadena          Par:          " +
subcadAsciiPar());
    cuadroSalida.println ("Subcadena          Impar:         " +
subcadAsciiImpar());
    cuadroSalida.println ("Número Vocales: " + contVocales());
    cuadroSalida.println ("Suma          Mayúsculas:          " +
sumCodigosAscii());
}

}

package septiembre;

class MiStringMain
{
    public static void main (String[] args)
    {
        MiString miCadena = new MiString( );
        miCadena.empezar();
    }
}

```