

1. (0.5 puntos) Indicar y justificar claramente cuál es el valor final de los datos miembro x, y, z.

```
class P1Main
{
    public static void main (String [] args)
    {
        Pregunta1 p1 = new Pregunta1();
        p1.empezar();
    }
}

import javabook.*;
class Pregunta1
{
    private int x,y,z;
    private MainWindow ventanaPrincipal;
    private OutputBox cuadroSalida;
    public void empezar()
    {
        int x,y;
        inicializar();
        x = y = 20;
        modificar(x,y);
        visualizar();
    }
    private void inicializar()
    {
        ventanaPrincipal = new MainWindow();
        cuadroSalida = new OutputBox(ventanaPrincipal);
        ventanaPrincipal.setVisible(true);
        cuadroSalida.setVisible(true);
        x = 200;
        y = 250;
        z = 300;
    }
    private void modificar(int x, int y)
    {
        z = x + y;
        x = z;
        y = 2 * z;
    }
    private void visualizar()
    {
        cuadroSalida.println("x= "+x+" y= "+y+" z= "+z);
    }
}
```

Solución:

x = 200 y = 250 z = 40

2. (1 punto) Escribir las sentencias necesarias para calcular la siguiente suma:

$$1 + 3 + 7 + 15 + 31 + \dots + (2^{20} - 1)$$

- a. Mediante un bucle for.
- b. Mediante un bucle do-while.

Solución:

a. Bucle for

```
double termino, suma = 1.0;
for (int i=2; i<=20; i++)
{
    termino = Math.pow(2,i) - 1.0;
    suma += termino;
}
```

b. Bucle do-while

```
double termino, suma = 1.0;
int i=2;
do
{
    termino = Math.pow(2,i) - 1.0;
    suma += termino;
    i++;
}while (i<=20);
```

3. (0.5 puntos) ¿Cuál es el resultado de la variable x en estos dos bloques de sentencias?

```
String s1 = new String ("Programación");
String s2 = new String ("en Vitoria");
s1 = s2;
boolean x = (s1 == s2);
```

```
String cad1 = new String ("Febrero");
String cad2 = cad1;
cad1 = cad1 + "y Marzo";
int x = cad2.length();
```

Solución:

x = true

x = 7

4. (3 puntos) Representar un número entero (en base 10) en una de las siguientes bases: 2, 8 ó 16.

- Los datos se introducen de forma correcta a través de la línea de comandos. Por ejemplo: BaseMain 1357 16, siendo 1357 el número entero y 16 la base.
- Para el ejemplo anterior la salida en pantalla debe mostrar:
El número 1357 en base 16 es: 54D
- El contenido de la clase BaseMain es el siguiente:

```
class BaseMain
{
    public static void main (String [] args)
    {
        String entrada1 = args[0];
        String entrada2 = args[1];
        int decimal = Integer.parseInt(entrada1);
        int base = Integer.parseInt(entrada2);
        Base decimalAOtraBase = new Base(decimal,base);
        decimalAOtraBase.visualizar();
    }
}
```

Se pide realizar el programa `Base.java`, considerando las siguientes condiciones:

- Sólo hay 2 atributos o datos miembro de tipo entero: decimal y base.
- El contenido parcial del método *visualizar* es:

```
public void visualizar()
{
    System.out.print("El numero " + ... + "\n");
}
```

- Los prototipos de los otros métodos a codificar son:

```
private String convertir()
private char letraHexa(int numero)
```

Solución:

```
class Base
{
    //datos miembro
    private int decimal;
    private int base;
    //constructor
    public Base(int decimal,int base)
    {
        this.decimal = decimal;
        this.base = base;
    }
    //métodos públicos
    public void visualizar()
    {
        System.out.print("El numero " + decimal + " en base "
            + base + " es: " + convertir() + "\n");
    }
    //métodos privados
    private String convertir()
    {
        int dividendo = decimal;
        int divisor = base;
        int resto;
        char letra;
        String cadena = "";
        while (dividendo>=base)
        {
            resto = dividendo%divisor;
            dividendo = dividendo/divisor;
            if(resto>9)
            {
                letra = letraHexa(resto);
                cadena = letra+cadena;
            }
            else
                cadena = resto+cadena;
        }
        if (dividendo>9)
            cadena = letraHexa(dividendo)+cadena;
        else
            cadena = dividendo+cadena;
        return cadena;
    }
}
```

```
private char letraHexa(int numero)
{
    char letra = ' ';
    switch (numero)
    {
        case 10:
            letra = 'A';
            break;
        case 11:
            letra = 'B';
            break;
        case 12:
            letra = 'C';
            break;
        case 13:
            letra = 'D';
            break;
        case 14:
            letra = 'E';
            break;
        case 15:
            letra = 'F';
            break;
    }
    return letra;
}
}
```

5. (5 puntos) En el concurso hipotético “El Gran Test de Programación I”, cuyo formato es una copia del que se emite cada domingo a las 10 de la noche, se evalúa la memoria a partir de las respuestas dadas a varias preguntas. Una de ellas consiste en acertar qué frase se esconde tras una frase original donde cada palabra está invertida. Por ejemplo, la frase de partida

em atsug avaj

corresponde a la frase de salida

me gusta java

Para evaluar vuestro conocimiento sobre la clase String, se pide realizar un programa en Java que proporcione la frase correcta. El programa se debe codificar utilizando las clases del paquete javabook y teniendo en cuenta las siguientes condiciones:

- El usuario puede teclear tantas frases como quiera.
- Si en la frase aparece algún dígito numérico, se considera incorrecta y se debe volver a introducir.
- Se supone que la frase original está en minúsculas y que finaliza en una letra. Cada palabra de la frase está separada por un único espacio en blanco.
- Se deben implementar dos clases denominadas FraseMain y Frase.
- Como mínimo se deben escribir estos dos métodos con el siguiente prototipo:

```
private boolean fraseEsCorrecta()
private String recomponerPalabra (String palabra)
```

- Se valorará el diseño de la clase Frase y el uso correcto de la clase StringBuffer, allí donde sea necesario.
- El contenido parcial de la clase Frase es:

```
public void empezar ()
{
    int respuesta;
    respuesta = cuadroRespuesta.prompt("¿Quieres obtener la frase correcta?");
    while (respuesta == cuadroRespuesta.YES)
    {
        leerFrase();
        obtenerFraseSalida();
        mostrarResultado();
        respuesta = cuadroRespuesta.prompt("¿Quieres obtener más frases?");
    }
}

private void leerFrase()
{
    fraseEntrada = cuadroEntrada.getString("Introducir la frase original: ");
    while(!fraseEsCorrecta())
    {
        cuadroMensaje.show("La frase contiene caracteres numéricos. Repetir. ");
        fraseEntrada = cuadroEntrada.getString("La frase original es: ");
    }
}
```

Solución:

```
import javabook.*;
```

```

class Frase
{
    /*******
        Datos Miembro
        *****/

    private MainWindow ventanaPrincipal;
    private InputBox cuadroEntrada;
    private OutputBox cuadroSalida;
    private ResponseBox cuadroRespuesta;
    private MessageBox cuadroMensaje;

    private String fraseEntrada,
                  fraseSalida;

    /*******
        Constructor
        *****/

    public Frase()
    {
        ventanaPrincipal = new MainWindow("Frases con palabras
invertidas");
        cuadroSalida = new OutputBox(ventanaPrincipal);
        cuadroRespuesta = new ResponseBox(ventanaPrincipal);
        cuadroEntrada = new InputBox(ventanaPrincipal);
        cuadroMensaje = new MessageBox(ventanaPrincipal);

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

    /*******
        Métodos Públicos
        *****/

    public void empezar ()
    {
        int respuesta;
        respuesta = cuadroRespuesta.prompt("¿Quieres obtener la
frase correcta?");
        while (respuesta == cuadroRespuesta.YES)
        {
            leerFrase();
            obtenerFraseSalida();
            mostrarResultado();
            respuesta = cuadroRespuesta.prompt("¿Quieres obtener
más frases?");
        }
    }

    /*******
        Métodos Privados
        *****/

    private void leerFrase()
    {

```

```

        fraseEntrada = cuadroEntrada.getString("Introducir la
frase original: ");
        while(!fraseEsCorrecta())
        {
            cuadroMensaje.show("La frase contiene caracteres
numéricos. Repetir. ");
            fraseEntrada = cuadroEntrada.getString("La frase
original es: ");
        }
    }

private boolean fraseEsCorrecta ()
{
    boolean esCorrecta = true;
    int i = 0;
    int longitud = fraseEntrada.length();
    while (i < longitud && esCorrecta)
    {
        char car = fraseEntrada.charAt(i);
        if (car >= '0' && car <= '9')
        {
            esCorrecta = false;
        }
        i++;
    }
    return esCorrecta;
}

private void obtenerFraseSalida()
{
    int i = 0;
    int longitud = fraseEntrada.length();
    int inicio, fin;
    String palabraEntrada, palabraSalida;
    fraseSalida = "";

    while (i < longitud)
    {
        inicio = i;
        while (i < longitud && fraseEntrada.charAt(i) != ' ')
        {
            i++;
        }

        fin = i;
        palabraEntrada = fraseEntrada.substring(inicio,fin);
        cuadroSalida.println(palabraEntrada);
        palabraSalida = recomponerPalabra(palabraEntrada);
        fraseSalida = fraseSalida + palabraSalida + " ";
        i++;
    }
}

private String recomponerPalabra(String palabra)
{
    StringBuffer palabraInver = new StringBuffer(palabra);
    return palabraInver.reverse().toString();
}

```

```
private void mostrarResultado()
{
    cuadroSalida.skipLine(3);
    cuadroSalida.println("Frase de Partida:");
    cuadroSalida.println("          " + fraseEntrada);
    cuadroSalida.skipLine(1);
    cuadroSalida.println("Frase de Salida:");
    cuadroSalida.println("          " + fraseSalida);
}
}
```

```
class FraseMain
{
    public static void main (String[] args)
    {
        Frase frase = new Frase();
        frase.empezar();
    }
}
```