



REPORTE DE PRÁCTICA 2

Aplicaciones

Juan Francisco Bielma Vargas

juan.bielma@gmail.com

29 de agosto de 2016

Introducción

Java

Java es un lenguaje de programación creado por James Gosling en 1991. El objetivo de Java es escribir un programa una vez y entonces correrlo en múltiples sistemas operativos.

Java está definido por una especificación y consiste en un lenguaje de programación, un compilador, núcleo, archivos de biblioteca y la máquina virtual de Java. La máquina virtual permite a los desarrolladores de software escribir código en otros lenguajes de programación que seguirán corriendo en la máquina virtual de Java

El lenguaje Java fue diseñado con las siguientes propiedades:

- Plataforma independiente: Los programas en Java usan la máquina virtual como abstracción y no acceden al sistema operativo directamente. Esto hace a los programas en Java portables. Un programa en Java puede correr sin modificarse en todas las plataformas soportadas.
- Lenguaje de programación orientado a objetos: Sin contar los tipos primitivos de datos, todos los elementos de Java son objetos.
- Lenguaje de programación fuertemente tipeado. Los tipos de las variables usadas deben estar predefinidas y la conversión a otros objetos es relativamente estricta, en la mayoría de los casos debe ser hecho por el programador.
- Lenguaje interpretado y compilado. El código fuente de Java es transferido al formato bytecode el cual no depende de la plataforma. Estas instrucciones bytecode son interpretadas por la máquina virtual.
- Administración de memoria automático. Java administra la asignación de la memoria y la de-asignación para crear nuevos objetos. El programa no tiene acceso directo a la memoria. El así llamado, recolector de basura, automáticamente elimina objetos cuales no existan punteros activos.

La sintaxis de Java es similar a C++. Java es sensible a mayúsculas y minúsculas. Ejemplo, las variables `miVariable` y `mivariable` son tratadas como diferentes variables.

Applet

Un Java Applet es un programa especial de Java que un navegador habilitado con tecnología Java puede descargar de internet y correr. Un applet es usualmente incrustado dentro de una página web y corre en el contexto de un navegador. Un applet debe ser una subclase de la clase `java.applet.Applet`. La clase Applet proporciona la interfaz estándar y el ambiente del navegador.

DESARROLLO

Error de desbordamineto

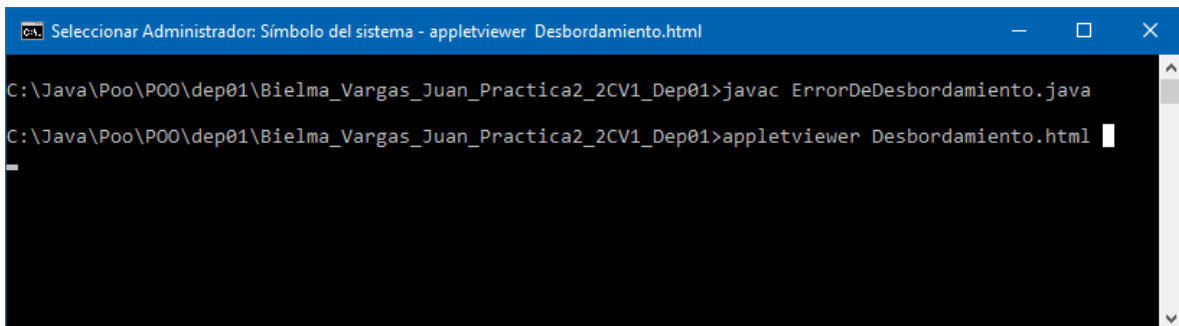
```
/**
 * La clase ErrorDeDesbordamiento implemente un applet
 * que muestra dos números, uno positivo y uno negativo
 * en pantalla
 */
import java.awt.*;
import java.applet.*;

public class ErrorDeDesbordamiento extends Applet
{
    private long positivo;
    private long negativo;

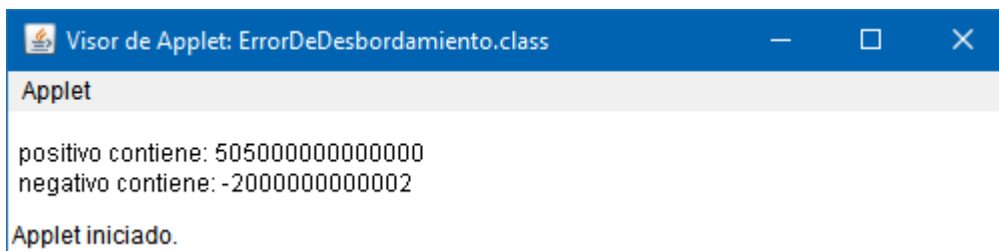
    public void init(){
        positivo = 5000000000000000L + 5000000000000000L;
        negativo = -2 - 2000000000000000L;
    }

    public void paint (Graphics g)
    {
        g.drawString("positivo contiene: " + positivo, 5, 25);
        g.drawString("negativo contiene: " + negativo, 5, 40);
    }
}
```

En las siguientes captura podremos ver que el código anterior fue compilado sin problemas y al iniciarlo en un Applet se ven los dos números.



```
C:\Java\Poo\P00\dep01\Bielma_Vargas_Juan_Practica2_2CV1_Dep01>javac ErrorDeDesbordamiento.java
C:\Java\Poo\P00\dep01\Bielma_Vargas_Juan_Practica2_2CV1_Dep01>appletviewer Desbordamiento.html
```



¿Cules son las características principales de un applet en JAVA?

- Las applets no tienen un método main() con el que comience la ejecución.
- Todas las applets derivan de la clase java.applet.Applet. Los applets deben redefinir ciertos métodos heredados de Applet que controlan su ejecución: init(), start(), stop(), destroy().
- Se heredan otros muchos métodos de las super-clases de Applet que tienen que ver con la generación de interfaces gráficas de usuario (AWT). Así, los métodos gráficos se heredan de Component, mientras que la capacidad de añadir componentes de interface de usuario se hereda de Container y de Panel.
- Las applets también suelen redefinir ciertos métodos gráficos: los más importantes son paint() y update(), heredados de Component y de Container; y repaint() heredado de Component.
- Las applets disponen de métodos relacionados con la obtención de información, como por ejemplo: getAppletInfo(), getAppletContext(), getParameterInfo(), getParameter(), getCodeBase(), etDocumentBase(), e isActive().

Dibuja la estructura básica de un Applet

```
import java.awt.Graphics;  
import java.applet.*;
```

```
public class EsqueletoApplet extends Applet {  
    public void init(){  
    }  
    public void start(){  
    }  
    public void stop(){  
    }  
    public void destroy(){  
    }  
    public void paint(Graphics g){  
    }  
}
```

Siguiendo la práctica nos pide que corriamos y pasemos el siguiente código al paradigma orientado a objetos

```
public Class HolaMundo  
{  
    public static void main (String arg[])  
    { System.out.println("Hola, Mundo");  
    }  
}
```

Este código lo que genera es imprimir Hola, Mundo en la consola.
Al llevarlo al paradigma orientado a objetos quedaría de la siguiente forma.

```
/**  
 * imprime a pantalla "Hola Mundo Java!" con la salida estandar.
```

```

*/
public class HolaMundoApp {
    public static void main(String[] args) {

        Mensaje mensaje = new Mensaje();
        mensaje.setMensaje("Hola Mundo Java!");
        mensaje.imprimir();

    }
}

```

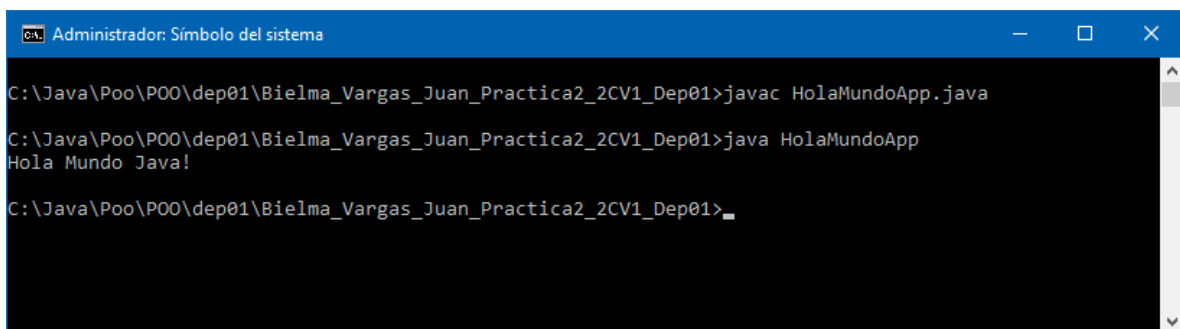
Podemos ver en el código anterior que requerimos de un objeto Mensaje por lo cual generamos la siguiente clase.

```

/**
 * La clase Mensaje tiene un atributo mensaje y un metodo que lo imprime
 */
public class Mensaje {
    private String mensaje;

    //metodo que imprime la variable msj
    public void imprimir(){
        System.out.println(mensaje);
    }
    //Getter para la variable msj
    public void setMensaje(String mensaje){
        this.mensaje = mensaje;
    }
}

```



```

C:\Java\Poo\P00\dep01\Bielma_Vargas_Juan_Practica2_2CV1_Dep01>javac HolaMundoApp.java

C:\Java\Poo\P00\dep01\Bielma_Vargas_Juan_Practica2_2CV1_Dep01>java HolaMundoApp
Hola Mundo Java!

C:\Java\Poo\P00\dep01\Bielma_Vargas_Juan_Practica2_2CV1_Dep01>_

```

¿Cuales son las principales diferencias estructurales entre este programa y el anterior?

R= En el primero se requiere de un entorno gráfico para imprimir la información mientras que en el segundo se imprime en consola.

¿En qué se diferencia un Applet de una aplicación?

R= En el método main, mientras que la aplicación siempre lo necesitará, una applet no.

Una ejemplo más.

Compila nuevamente el siguiente programa y corrige los errores, así como la indexación:

```
class prueba2
{
Public static void main (string args[])
{
system.outprintln("\007\007\007Hola Mundo nuevamente" + "en
todos los planetas");
}
}
```

Al corregirlo y pasarlo a orientado a objetos quedaría de la siguiente manera.

```
public class HolaPlanetas
{
    public static void main (String[] args)
    {
        Mensaje mensaje = new Mensaje();
        mensaje.setMensaje("\007\007\007Hola Mundo nuevamente en todos los planetas");
        mensaje.imprimir();
    }
}
```

¡A quebrarse la cabeza!:

Código en la clase principal Chequera.java

```
import java.io.*;

public class Chequera {
    public static void main (String args[]) throws IOException {
        Cheque cheque = new Cheque();
        cheque.leerInformacion();
        cheque.imprimir();
    }
}
```

Clase Cheque

```
import java.io.*;
import java.util.*;
import java.text.DateFormat;
import java.text.SimpleDateFormat;

public class Cheque
{

    private String nombre;
```

```

private double cantidad;
private String banco;
private Date fecha;
private String fechaFormateada;
private static DateFormatter dateFormatter = new SimpleDateFormat("dd-MMM-yyyy");

public Cheque(){
    nombre = "Juan Bielma";
    cantidad = 10;
    banco = "Banco Patito de Mexico";
    fecha = new Date();
    fechaFormateada = dateFormatter.format(fecha);
}

public void setNombre(String nombre){
    this.nombre = nombre;
}

public String getNombre(){
    return this.nombre;
}

public void setCantidad(double cantidad){
    this.cantidad = cantidad;
}

public double getCantidad(){
    return cantidad;
}

public String getBanco(){
    return banco;
}

public Date getFecha(){
    return fecha;
}

public String getFechaFormateada(){
    return fechaFormateada;
}

public void leerInformacion() throws IOException
{
    String entradaUsuario = "";
    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

    System.out.println("Dame tu nombre: ");
    nombre = br.readLine();
    System.out.println("Dame la cantidad: ");
}

```

```

while (!Validacion.isNumeric(entradaUsuario)){
    entradaUsuario = br.readLine();
    if (Validacion.isNumeric(entradaUsuario))
        cantidad = Double.parseDouble(entradaUsuario);
    else
        System.out.println ("Entrada no válida, no introdujo un número");

}

}

public void imprimir(){

    System.out.print("\033[H\033[2J");
    System.out.println("\n" + banco + "\t\t\t\t\t\t\tFecha:" + fechaFormateada);
    System.out.println("Nombre: " + nombre);
    System.out.println("Cantidad c/letra: " + NumeroPalabra.convertir(cantidad) + "
\t$" + cantidad );

    System.out.println("\nFirma:_____");
}

}

```

Clase NumeroPalabra.java

```

import java.text.DecimalFormat;

public class NumeroPalabra {

    private static final String[] decenas = {
        "",
        " diez",
        " veinte",
        " treinta",
        " cuarenta",
        " cincuenta",
        " sesenta",
        " setenta",
        " ochenta",
        " noventa"
    };

    private static final String[] numeros = {
        "",
        " uno",
        " dos",
        " tres",

```



```

" cuatro",
" cinco",
" seis",
" siete",
" ocho",
" nueve",
" diez",
" once",
" doce",
" trece",
" catorce",
" quince",
" dieciseis",
" diecisiete",
" dieciocho",
" diecinueve",
" veinte",
" veintiuno",
" veintidos",
" veintitres",
" venticuatro",
" venticinco",
" vientosis",
" vientosiete",
" vientosiocho",
" veintinueve",
};

```

```

private NumeroPalabra() {}

```

```

private static String convertirMenorAMil(int numero) {
    String numeroPalabra = "";

```

```

    if (numero % 100 < 30){
        numeroPalabra = numeros[numero % 100];
        numero /= 100;
    }
    else {

```

```

        numeroPalabra = numeros[numero % 10];

```

```

        numero /= 10;

```

```

        numeroPalabra = numeroPalabra == "" ? decenas[numero % 10] : decenas[numero %
10] + " y " + numeroPalabra;
        numero /= 10;
    }
    if (numero == 0) return numeroPalabra;
    if(numero == 1) return numeroPalabra == "" ? "cien" : "ciento" + numeroPalabra;

```

```

    if(numero == 5) return "quinientos" + numeroPalabra;
    if(numero == 7) return "setecientos" + numeroPalabra;
    if(numero == 9) return "novecientos" + numeroPalabra;
    return numeros[numero] + "cientos" + numeroPalabra;
}

```

```

public static String convertir(double numero) {

    long pesos = (long)Math.floor( numero ) ;
    long centavos = (long)(Math.round(-1f * (numero - pesos ) * 100f) * -1f);

    return (convertir(pesos) + " punto " + convertir(centavos) + " MXN").toUpperCase();

}

```

```

public static String convertir(long numero){
    // 0 to 999 999 999 999
    if (numero == 0) { return "zero"; }

    String snumero = Long.toString(numero);

    // pad with "0"
    String mask = "0000000000000";
    DecimalFormat df = new DecimalFormat(mask);
    snumero = df.format(numero);

    // XXXnnnnnnnnnn
    int billions = Integer.parseInt(snumero.substring(0,3));
    // nnnXXXnnnnnnn
    int millions = Integer.parseInt(snumero.substring(3,6));
    // nnnnnnXXXnnn
    int hundredThousands = Integer.parseInt(snumero.substring(6,9));
    // nnnnnnnnnXXX
    int thousands = Integer.parseInt(snumero.substring(9,12));

```

```

String tradBillions;
switch (billions) {
case 0:
    tradBillions = "";
    break;
case 1 :
    tradBillions = convertirMenorAMil(billions)
    + " un billon ";
    break;
default :
    tradBillions = convertirMenorAMil(billions)
    + " billones ";
}

```

```

String result = tradBillions;

String tradMillions;
switch (millions) {
case 0:
    tradMillions = "";
    break;
case 1 :
    tradMillions = convertirMenorAMil(millions)
        + " un millon ";
    break;
default :
    tradMillions = convertirMenorAMil(millions)
        + " millones ";
}
result = result + tradMillions;

String tradHundredThousands;
switch (hundredThousands) {
case 0:
    tradHundredThousands = "";
    break;
case 1 :
    tradHundredThousands = " mil ";
    break;
default :
    tradHundredThousands = convertirMenorAMil(hundredThousands)
        + " mil ";
}
result = result + tradHundredThousands;

String tradThousand;
tradThousand = convertirMenorAMil(thousands);
result = result + tradThousand;

// remove extra spaces!
return result.replaceAll("^\\s+", "").replaceAll("\\b\\s{2,}\\b", " ");
}

}

```

Clase validacion.java

```

public class Validacion{

    public static boolean isNumeric(String str) {
        return (str.matches("[+-]?\\d*(\\.\\d+)?") && str.equals(""))==false;
    }

}

```

```
Administrador: Símbolo del sistema - java Chequera
C:\Java\Poo\P00\dep01\Bielma_Vargas_Juan_Practica2_2CV1_Dep01>javac Chequera.java
C:\Java\Poo\P00\dep01\Bielma_Vargas_Juan_Practica2_2CV1_Dep01>java Chequera
Dame tu nombre:
Juan Bielma
Dame la cantidad:
361868131815.16
```

```
Seleccionar Administrador: Símbolo del sistema
Banco Patito de Mexico                               Fecha: 29-ago-2016
Nombre: Juan Bielma
Cantidad c/letra: TRESCIENTOS SESENTA Y UNO BILLONES OCHOCIENTOS SESENTA Y OCHO MILLONES CIENTO TRE
INTA Y UNO MIL OCHOCIENTOS QUINCE PUNTO DIECISEIS MXN  $3.6186813181516E11
Firma: _____
C:\Java\Poo\P00\dep01\Bielma_Vargas_Juan_Practica2_2CV1_Dep01>
```

1. ¿Cuál fue el error más común en los anteriores programas?

R= Principalmente fueron errores de sintaxis en el código

2. ¿Cuánto tiempo invertiste en encontrar los errores?

R= Bastante, me llevo un tiempo definir las palabras que estaban mal escritas y corregirlas.

Conclusiones

Al principio fue complicado el cambio de paradigma y encontrar los errores en el código dado, ya que algunas de ellas eran de sintaxis o que un carácter este en mayúsculas cuando debería estar en minúsculas o viceversa.

BIBLIOGRAFÍA 1.

<http://nereida.deioc.ull.es/~cleon/hli00/slides/tsld011.htm>

<http://www.vogella.com/tutorials/JavaIntroduction/article.html>

<http://java-desd-cero.blogspot.mx/2012/06/applets-que-es-un-applets.html>