Java. Práctica 1

NOTA

En esta primera práctica no vamos a utilizar ningún entorno de desarrollo sino simplemente los comandos que proporciona el JDK1.3 y las herramientas de Windows.

- Abrir una sesión MS-DOS y crear el directorio `c:\java`. A partir de este directorio iremos creando directorios con las clases nuestras.
- Por otro lado, abrir el bloc de notas. Aquí escribiremos nuestros programas. A la hora de ponerle un nombre al fichero, hacedlo entrecomillado; por ejemplo "Urna.java" (Esto impide que el bloc de notas ponga su extensión por defecto que es .txt)

EDICIÓN: Lo haremos con el bloc de notas de Windows.
COMPILACIÓN: Desde la sesión MS-DOS abierta.
EJECUCIÓN: Desde la sesión MS-DOS abierta.

Ejercicio1. Directorio de trabajo: `c:\java\hola`. Realizar una prueba de edición, compilación y ejecución con un programa mínimo que escriba "Hola Mundo".

Ejercicio 2. Directorio de trabajo: `c:\java\punto`. Crear la clase `Punto` y un programa que utilice dos puntos para calcular su distancia. Crear así mismo, la clase `Píxel`.

Ejercicio 3. Directorio de trabajo: `c:\java\urnas`. Crear la clase `Urna`. Realizar dos versiones de un programa que utilice una urna. La primera crea una urna conocidos el número de bolas blancas y negras en tiempo de compilación. En la segunda estos datos se conocen en tiempo de ejecución. Probar con distinto número de bolas y analizar los resultados obtenidos hasta llegar a alguna conclusión.

Ejercicio 4. Directorio de trabajo: `c:\java\primos`. Pretendemos generar una lista enlazada de números primos desde el 2 hasta un número dado como tope. Para ello:
Construimos inicialmente una celda que contiene: al número 2 y una referencia (inicialmente nula) a otra posible celda. Posteriormente, partiendo de la celda que contiene al 2, iteramos desde el 3 hasta el número dado como tope realizando lo siguiente:

Si el número en cuestión es múltiplo del que contiene la celda no hacemos nada.
Si no es múltiplo, vemos si esa celda tiene referenciada a otra celda.
Si la tiene referenciada, continuamos el proceso con esta nueva celda.
Si no la tiene referenciada, creamos una celda que contiene al número con el que iteramos y la referenciamos como siguiente celda a la actual.

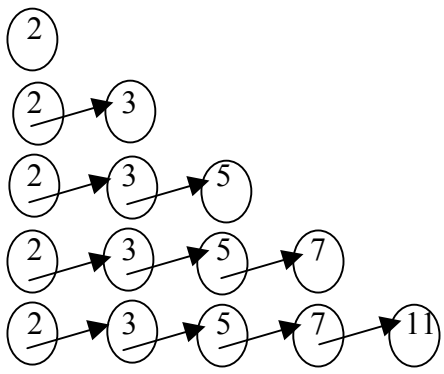
3,4,5,6,7,8,...., tope

4,5,6,7,8,...., tope

6,7,8,...., tope

8,9,10,11,12,...., tope

12,...., tope



Realizar la clase `Primo` que implemente el comportamiento de la celda.
Realizar un programa que permita generar la lista de valores primos hasta un tope dado. Este programa deberá después recorrer la lista enlazada de primos y mostrarlos por la pantalla.

Ejercicio 5. Directorio de trabajo: `c:\java\listaint`. Crear la clase `ListaInt` que permita generar una lista de valores enteros. El protocolo debe ser el siguiente:

```
// Cierto si la lista está vacía
public boolean isEmpty()
// Añade un entero por la cabeza de la lista
public void addFirst(int n)
// Igual que addFirts
public void add(int n)
// Añade un elemento por el final de la lista
public void addLast(int n)
// Devuelve el primero elemento de la lista
public int getFirst()
// Devuelve el último elemento de la lista
public int getLast()
// Devuelve y elimina el primer elemento de la lista
public int removeFirst()
// Devuelve y elimina el último elemento de la lista
public int removeLast()
// Crea un String que representa a la lista en su estado actual
public String toString()
```

Para la creación de esta clase se debe crear otra que llamaremos `NodoListaInt` que representa a un nodo de la lista de enteros.
Crear una aplicación que utilice manipule objeto de esta lista mostrando los distintos estados por los que pasa.

Ejercicio 6. Directorio de trabajo: `c:\java\listaint`. Crear la clase `ColaInt` que, basándose en la clase del ejercicio 4, implemente el protocolo de una cola de enteros:

```
// Cierto si la cola está vacía
public boolean isEmpty()
// Añade un entero a la cola
public void add(int n)
// Devuelve y elimina el siguiente elemento de la cola
public int get()
// Crea un String que representa a la cola en su estado actual
public String toString()
```

Ejercicio 7. Directorio de trabajo: `c:\java\listaint`. Crear la clase `PilaInt` que, basándose en la clase del ejercicio 4, implemente el protocolo de una pila de enteros:

```
// Cierto si la pila está vacía
public boolean isEmpty()
// Añade un entero a la cola
public void push(int n)
// Devuelve el siguiente elemento de la pila
public int top()
// Devuelve y elimina el siguiente elemento de la pila
public int pop()
// Crea un String que representa a la pila en su estado actual
public String toString()
```