

Práctica 3: Fechas

1. Objetivo

El objetivo de la práctica es la elaboración de un programa (Fechas) que permita al usuario realizar algunos cálculos con fechas. Para ello, habrá que poner en práctica cuestiones sobre:

- Estructuras alternativas o condicionales.
- Estructuras repetitivas.
- Manejo básico de excepciones. para validar las entradas del usuario.
- Programación modular

Además, a través de la ampliación se pueden trabajar otros aspectos:

- la ruptura de un string en trozos o “tokens” usando StringTokenizer

2. Introducción.

El programa permitirá al usuario realizar tres acciones entre las que podrá elegir a través de un menú:

1.- Validar una fecha: Se solicitará al usuario una fecha y el programa le dirá si es válida o inválida.

2.- Calcular el número de días que hay entre dos fechas: Se solicitará al usuario dos fechas y el programa le dirá cuántos días de diferencia hay entre ambas.

3. El menú.

El programa mostrará un menú con aspecto similar al siguiente:

- 1.- Validar fecha.
- 2.- Calcular diferencia.
- 0.- Salir

Indique la acción a realizar:

- Si el usuario elige una opción correcta (1 o 2), el programa solicitará la información necesaria para llevar a cabo la acción elegida, según se ha descrito en el apartado anterior. Se mostrarán los resultados y se mostrará de nuevo el menú.
- Si el usuario elige salir (0), el programa terminará.

- Si el usuario se equivoca e indica una opción incorrecta, el programa le informará del error y se mostrará de nuevo el menú.

4. Validar una fecha (opción 1 del menú).

Cuando el usuario elije la primera opción del menú, se le solicita la introducción de una fecha. El usuario debe introducir una fecha con el aspecto "dd/mm/aaaa". (dd, mm y aaaa pueden tener longitud variable, por ejemplo 1/1/2009 , 31/12/2009, etc).

El programa determinará si la fecha introducida es o no válida. En caso de ser correcta, se mostrará la fecha con el siguiente formato:

*dia de **nombreDelMes** de año: Fecha correcta.*

Por ejemplo:

*1 de **enero** de 2009: Fecha correcta*

Es decir, se mostrará el **nombre del mes** en lugar del número del mes, que es lo que el usuario ha introducido.

- Si la fecha es incorrecta, se indicará al usuario. Por ejemplo:

Día: 30, Mes: 2 Año: 2012. Fecha incorrecta

Consideraciones para determinar si la fecha es o no válida.

Para que una fecha sea válida tiene que cumplir una serie de condicionantes:

- Se tiene que haber introducido con el formato que se indica. Por ejemplo "11_10_2000" se considera una fecha inválida porque el formato con el que se ha expresado no es el requerido.
- El año tiene que ser positivo.
- El mes tiene que estar entre 1 y 12
- El día, debe ser uno de los que tiene el mes, teniendo en cuenta que:
 - Hay meses con 30 y meses con 31 días.
 - Febrero puede tener 29 o 28 días, dependiendo de si el año es o no bisiesto.

5. Calcular diferencia (opción 2 del menú).

Cuando el usuario elije la segunda opción del menú, se le solicita la introducción de dos fechas y se muestra al usuario cuántos días de diferencia hay entre una y otra.

Si el usuario introduce mal alguna fecha (formato incorrecto o fecha incorrecta), la fecha mal introducida se le volverá a solicitar.

El número de días que se mostrarán serán los correspondientes a la diferencia entre la segunda fecha y la primera. Así, si la primera es menor, el resultado será positivo mientras que si la menor es la segunda, el resultado será negativo.

6. Descomposición modular del programa

Para facilitar la resolución, es muy importante que el programa se descomponga en módulos, creando métodos que realicen tareas o cálculos de utilidad:

Antes de empezar a escribir líneas de programa:

- piensa qué métodos te pueden resultar útiles.
- Piensa en qué problemas tendrás que resolver y trata de descomponerlos en subproblemas.

7. Consideraciones importantes

- No se pueden usar funciones de fechas que estén ya disponibles en el lenguaje Java: Java dispone de clases que permiten realizar cálculos con fechas. No se pueden utilizar
- El programa debe ser robusto en todo momento y no debe terminar de forma abrupta ante entradas inadecuadas del usuario.

8. Ampliaciones

a. StringTokenizer

Cuando el usuario introduce una fecha con la forma “dd/mm/aaaa”, extraemos de la cadena tres datos numéricos: el día, el mes y el año. Para ello utilizamos métodos de la clase String (charAt, substring, ...)

Java dispone de una clase *java.util.StringTokenizer* que permite “romper” un String en trozos, tomando como separador un carácter determinado.

Infórmate sobre *StringTokenizer* y modifica el programa para que la extracción del día, mes y año se realice utilizando esta clase.

Ten en cuenta si los métodos de StringTokenizer que utilizas lanzan alguna excepción .

b. Sumar días a una fecha.

Añadir al programa una tercera opción de menú “3.- Sumar días”.

Se trata de que el usuario introduzca una fecha y un número de días y se le muestre la fecha correspondiente a la original después de añadirle el número de días indicados.

9. Proyecto y entrega

La práctica se hará en un proyecto de nombre “proApellidoNombre.practica3Fechas”. La clase que contiene al método main se llamará “Fechas”

El proyecto se comprimirá en un archivo y se subirá a moodle.