Práctica 4:

Datos reales

Objetivo de la práctica

El objetivo de esta práctica es consolidar la utilización de los esquemas de composición algorítmica ejercitados en las prácticas anteriores y familiarizarse con los tipos **float** y **double**.

Ejercicios de la sesión práctica obligatorios (sobre 7 puntos)

1. Escribe un programa Java que escriba por pantalla el valor de la función sen(x) de un valor x introducido por teclado utilizando la serie:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!}$$

   Calcula la serie hasta que el termino calculado sea menor que la precisión deseada. Declara la precisión deseada como una constante (p.e. final double PRECISION = 1E-5;)

   Comprueba que tu implementación funciona correctamente mostrando el resultado de la serie y el resultado obtenido con la función Math.sin(x).

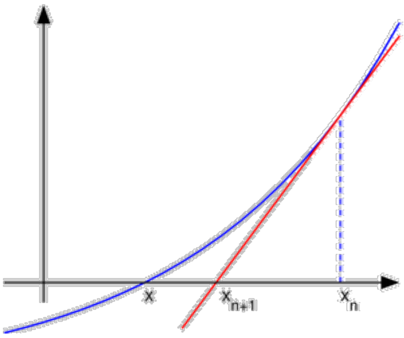
   Tu programa debe leer el valor de la x en grados, pero tanto en la serie como en la función Math.cos(x) y Math.sin(x) los valores de x deben estar en radianes. Para hacer la conversión puedes utilizar la constante Math.PI. Ejemplo de ejecución:

```
Dame valor de x:70
sin(1.2217304763960306)=0.9396926207859084
Calculado con Math.sin(1.2217304763960306)=0.9396926207859083
```

2. Utiliza el método de Newton para obtener las raíces de una ecuación de segundo grado. El método de Newton nos dice que, si X es una aproximación al cero de la función, x-(f(x)/f'(x)) es una aproximación mejor. Obtén una de las raíces de la ecuación de segundo grado ax^2-bx+c=0 por el método de Newton, y la otra sabiendo que $x_1 \cdot x_2 = c/a$. Utiliza los valores a = 1.0, b=1*10^5, c = -1.0 y distintas precisiones del resultado (p.e. precision=1E-5).

Método de Newton-Raphson

Para encontrar las raíces de una ecuación (Isaac Newton 1642-1727)



si X_n es una estimación de la raíz, X_{n+1} es una estimación mejor siendo f ' la derivada de la función en X_n.

tangente=$f(X_n)/(X_n-X_{n+1})$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

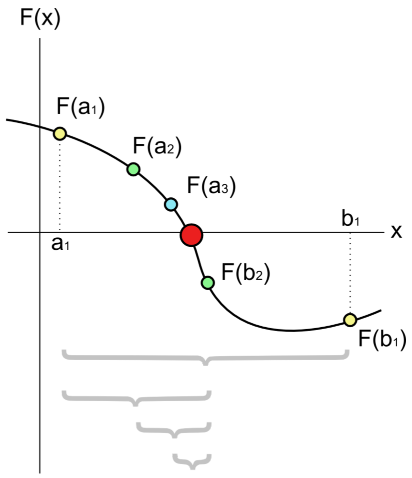
Método aproximado. Dejamos de iterar hasta que

|Estimación actual –Estimación previa| < precisión

Ejercicios de la sesión práctica voluntarios (1.5 puntos cada uno)

3. Utiliza el método de la bisección para obtener las raíces de una ecuación de segundo grado. Obtén una de las raíces por éste método descrito en http://es.wikipedia.org/wiki/Método_de_bisección, y la otra raíz sabiendo que $x_1 \cdot x_2 = c/a$. Utiliza los valores a = 1.0, b=1*10^5, c = -1.0. Utiliza un intervalo adecuado para encontrar una de las raices.

Método de la bisección



Supiendo que hay continuidad de la función *f*(*x*) en el intervalo [*a,b*], se verifica que f(a).f(b)< 0 (Hay cambio de signo en el intervalo).

Se calcula el punto medio *m* del intervalo [*a,b*] y se evalúa *f*(*m*) si ese valor es igual a cero, ya hemos encontrado la raíz buscada

En caso de que no lo sea, verificamos si *f*(*m*) tiene signo opuesto con *f*(*a*) o con *f*(*b*). Se redefine el intervalo [*a, b*] como [*a, m*] o [*m, b*] según se haya determinado en cuál de estos intervalos ocurre un cambio de signo

Con este nuevo intervalo se continúa sucesivamente encerrando la solución en un intervalo cada vez más pequeño, hasta alcanzar la precisión deseada

4. Escribe un programa Java que escriba por pantalla el valor de la función sin(x) de un valor x introducido por teclado utilizando la serie:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!}$$

Calcula la serie hasta que el termino calculado sea menor que la precisión deseada. Declara la precisión deseada como una constante (p.e. final double PRECISION = 1E-5;)

Comprueba que tu implementación funciona correctamente mostrando el resultado de la serie y el resultado obtenido con la función Math.sin(x). Ejemplo de ejecución:

```
Dame valor de x:180
   cos(3.141592653589793)=-1.0
   Calculado con Math.cos(3.141592653589793)=-1.0
```

La clase **Math**[1] representa la librería matemática de Java. La clase ofrece métodos estáticos (pertenecen a la clase, no a las instancias), por lo que se pueden invocar haciendo referencia a la clase, no es necesario crear instancias. Compara tus resultados con Math.**exp**() y Math.**sqrt**().

[1] http://download.oracle.com/javase/1.4.2/docs/api/java/lang/Math.html