Entrega 05

Carlos Gerardo Herrera Cortina – A00821946

**Diseño**

**Clase Simpson:**

#include “CalculosF.h”

#include <iostream>

Private:

Double x, dof;

Public:

Simpson (double x, double dof); // constructor que recibe los valores de x y dof introducidos por el usuario

Double calcSimpson(); // hace el calculo de p mientras llama al método getF(x) desde un objeto de clase CalculosF. Al final regresa el valor de p.

Void setXDOF(double x, double dof); // actualiza el valor de x y de dof del objeto creado

**Clase CalculosF**

#include <math.h>

Using namespace std;

Private:

Double x, dof;

Public:

CalculosF(double dof, double x); // constructor que recibe valores de dof y x introducidos en la clase Simpson.

Double getF(double x); // regresa F(x). Llama a getGamma(x) para hacer el calculo

Double getGamma(double x); // regresa a getGammaInt(x) o getGammaDouble(x) dependiendo del caso

Double getGammaInt(double x); // regresa el factorial de x-1 de forma recursiva

Double getGammaDoubel(double x); // regresa el calculo de gamma, haciendo la operación de manera recursiva

**MAIN:**

**#include “Simpson.h”**

* Inicializa variables double p, pAux, pUser y dof en 0.0.
* Inicializa variable double x en 1.0.
* Inicializa variable double d en 0.5
* Inicializa variable bool isLow en false.
* Pide p y se recibe del teclado
* Si hay un error de caracter o un número no válido, se muestra error y termina el programa
* Pide dof y se recibe del teclado
* Si hay un error de carácter se muestra error y termina el programa
* Si dof es un número correcto (significa que los 2 valores x y dof son correctos), se crea un objeto de clase Simpson con x y dof como parámetros de entrada del constructor. Después se llama a la función calcSimpson(), la cual hará el cálculo de p y se le asignará a Paux.
  + Si pAux > pUser, entonces isLow = false y se le resta a x el valor de d (x-=d). Si no es así, isLow = true y se le suma d a la variable x.
  + Se inicializa un ciclo (do while). Se seguirá ejecutando mientras el absoluto de la resta p – pAux > 0.00000001. El ciclo consiste en lo siguiente:
    - Se actualiza x y dof del objeto con método setXDOF(x, dof).
    - P recibe el valor de pAux, para que este ultimo regrese un nuevo calculo de calcSimpson.
    - Se hace una comparación:

Si pAux > pUser entonces isLow = false. Si la variable isLow era verdadera entonces d se divide entre dos.

Si pAux < pUser entonces isLow = true. Si la variable isLow era falsa entonces d se divide entre dos.

* + - Se hace otra comparación:

Si pAux < pUser sentonces a x se le suma el valor actual de d.

Si no es así, a x se le resta el valor de d.

* + Una vez termina el ciclo, se imprime la p del usuario con 5 decimales de precisión, dof especificado, y valor de x actual con 5 decimales de precisión.
* Si dof no es un número correcto, muestra error y termina programa.
* Termina el programa.

**Pruebas:**

* Las tres pruebas de los requerimientos, más 1 agregadas:
  + Teclear 0.2 6
  + Teclear 0.45 15
  + Teclear 0.495 4
  + Teclear 0.3 7 (caso añadido por mi)
* Casos de excepción: cuando hay un número incorrecto en p (ingresar 0.6) o probar con un dato de dof que tiene letras (ingresar 0.2 4a )