Functional Specification Template

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Student** | | | Alvaro Andres Suarez Alfonso | | **Date** | 01 Mar 2015 |
| **Program** | | | Tarea 6 | | **Program #** | CSOF5101\_01\_6 |
| **Instructor** | | | Luis Daniel Benavides Navarro | | **Language** | JAVA |
|  | | | | | | |
| **Class Name** | | Distribution | | | | |
| **Parent Class** | |  | | | | |
|  | | | | | | |
| **Attributes** | | | | | | |
|  | **Declaration** | | | **Description** | | |
|  | limitei :double | | | Limite inferior de la integral | | |
|  | limitef :double | | | Limite superior de la integral | | |
|  | dof :double | | | Grados de libertad | | |
|  | w :double | | | Adiciones por iteracion: limite superior / num\_seg | | |
|  | Num\_seg :double | | | Numero de segmentos | | |
|  |  | | |  | | |
|  | | | | | | |
| **Items** | | | | | | |
|  | **Declaration** | | | **Description** | | |
|  | Distribution(double limitei,double limitef,double dof, double num\_seg) | | | Constructor | | |
|  | Distribution() | | | Constructor generic | | |
|  | double truncDouble(double d) | | | Trunca un double en el formato ###0.## | | |
|  | double getLimitei() | | | Retorna limite inferior de la integral | | |
|  | double getLimitef() | | | Retorna limite superior de la integral | | |
|  | double getDof() | | | Retorna grados de libertad | | |
|  | double getW() | | | Retorna w | | |
|  | double getNum\_seg() | | | Retorna num\_seg | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **Class Name** | | TablaDatos | |
| **Parent Class** | |  | |
|  | | | |
| **Attributes** | | | |
|  | **Declaration** | | **Description** |
|  | static double ACCEPTABLEERROR | | Constante de error acceptable |
|  | static double NUM\_SEG | | Constante de num\_seg |
|  | static double LIMITEI | | Constante de limite inferior de la integral |
|  | List<TablaDatos> tablaDatos | | Lista de su mismo tipo |
|  | Distribution d | | Instancia de la clase distribution |
|  | int i | | Indice i |
| double xi | | Limite inferior de la integral |
| double fX | | Valor de la function f(x) |
| int multiplier | | Multiplicador |
| double terms | | Dato de Formula w\*multiplier\*f(x) |
| double p | | Valor de la integral |
|  | | | |
| **Items** | | | |
|  | **Declaration** | | **Description** |
|  | TablaDatos(double limitei,double limitef,double dof, double num\_seg) | | Constructor |
|  | TablaDatos(int i,double xi,int multiplier) | | Constructor de fila para TablaDatos |
|  | llenarTabladatos1(double limitei,double limitef,double dof, double num\_seg) | | Llena la tabla con los datos a buscar |
|  | calculadas() | | Calcula la integral |
|  | int esPar(int n) | | Retorna 2 si es par en caso contrario retorn 4 |
|  | truncDouble(double d) | | Trunca un double en el formato ###0.##### |
|  | double factorial(double d) | | factorial de un entero |
|  | double la\_gamma(double x) | | factorial de un fraccionario o entero |
|  | double getfX() | | Obtiene fX |
|  | double getTerms() | | Obtiene Terms |
|  | double getP() | | Obtiene p |
|  | List<TablaDatos> getTablaDatos() | | Obtiene tablaDatos |
|  | Distribution getD() | | Obtiene d |
|  |  | |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **Class Name** | | XCalculador | |
| **Parent Class** | |  | |
|  | | | |
| **Attributes** | | | |
|  | **Declaration** | | **Description** |
|  | static double ACCEPTABLEERROR | | Constante de error acceptable |
|  | static double NUM\_SEG | | Constante de num\_seg |
|  | static double LIMITEI | | Constante de limite inferior de la integral |
|  | double limiteF | | Valor buscado para limite superior |
|  | Double p | | Valor a encontrar para integral |
|  |  | |  |
|  | | | |
| **Items** | | | |
|  | **Declaration** | | **Description** |
|  | XCalculador(int mostrar) | | Constructor |
|  | void findX(double limitef,double dof,double pToGet) | | Busca valor superior de la integral |
|  | double getLimiteF() | | Retorna valor encontrado superior de la integral |
|  |  | |  |
|  |  | |  |

Functional Specification Template Instructions

|  |  |
| --- | --- |
| Purpose | * To hold a part’s functional specifications * To describe classes, program modules, or entire programs |
| General | * Use this template for complete programs, subsystems, or systems. * Use this template to document the functional specifications during planning, design, test development, implementation, and test. * After implementation and testing, update the template to reflect the actual implemented product. |
| Header | * Enter your name and the date. * Enter the program name and number. * Enter the instructor’s name and the programming language you are using. |
| Class Name | * Enter the part or class name and the classes from which it directly inherits. * List the class names starting with the most immediate. * Where practical, list the full inheritance hierarchy. |
| Attributes | * Provide the declaration and description for each global or externally visible variable or parameter with any constraints. * List pertinent relationships of this part with other parts together with the multiplicity and constraints. |
| Items | * Provide the declaration and description for each item. * Precisely describe the conditions that govern each item’s return values. * Describe any initialization or other key item responsibilities. |
| Example Items | An item could be a class method, procedure, function, or database query, for example. |