# Problem identification and requirements analysis

## Case Study: Bisection method

|  |  |
| --- | --- |
| Customer | Teacher |
| User | People who wish to use the program. |
| Functional requirements | * RF0- Bisection method. * RF1- Absolute value method. * RF2- Power method. * RF3- Factorial method. * RF4- Cosine method. |
| Context of the problem | A Java language program is desired that implements the bisection method, an iterative method that is used to find the root, or solution, bisects on an interval (containing the real root). The program will only calculate 3 functions, which the user must be able to choose:        The program must ask the user for the root search interval and if there is no real root in the entered interval, an error message must be displayed informing about it. To calculate functions 1 and 2, the program will work with the approximate definition of the cosine function (Taylor series): |
| non-functional requirements | * RNF0: The program must implement the operations necessary for the calculations, through methods or subroutines, without making use of the Java Math API or any other library that has implemented mathematical functions. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| I identifier and name | RF0- Bisection method | | | |
| Summary | The program should allow the user to choose between 3 functions to find the root, or solution, bisects in an interval (containing the root). The program must ask the user for a search interval that encapsulates the real root of the chosen function and then calculate the root, making use of the necessary operations to calculate said function (factorial, power, absolute value and cosine). If there is an error, the program must show it on the screen, otherwise it must show the result on the screen. | | | |
| Inputs | **input name** | **Datatype** | | **Condition valid values** |
| option | int | | Must be one of the available options |
| a | double | | Must be the first endpoint of the interval |
| b | double | | Must be the second endpoint of the interval |
| Result or Postcondition | The program calculates the root, or solution, bisects at the interval entered using the bisection method and displays the solution. | | | |
| Outputs | **output name** | | **Datatype** | **Format** |
| invalidRange | | string | Characters (Text) |
| c | | double | N/A |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| identifier and name | RF1- Absolute value method | | | |
| Summary | The program must allow to calculate the absolute value of a entered number and return the result. | | | |
| Inputs | **input name** | **Datatype** | | **Condition valid values** |
| num | double | | Must be the number to calculate |
| Result or Postcondition | The program calculates the absolute value of the entered number and returns the result. | | | |
| Outputs | **output name** | | **Datatype** | **Format** |
| num | | double | N/A |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| identifier and name | RF2- Power method | | | |
| Summary | The program must allow calculating the power of a number entered and return the result. | | | |
| Inputs | **input name** | **Datatype** | | **Condition valid values** |
| exp | int | | must be the exponent |
| base | double | | It must be the base |
| Result or Postcondition | The program calculates the power with the entered exponent and base and returns the result. | | | |
| Outputs | **output name** | | **Datatype** | **Format** |
| pow | | double | N/A |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| identifier and name | RF3- Factorial method | | | |
| Summary | The program must allow calculating the factorial of a entered number and return the result. | | | |
| Inputs | **input name** | **Datatype** | | **Condition valid values** |
| num | double | | Must be the number to calculate |
| Result or Postcondition | The program calculates the factorial of the entered number and returns the result. | | | |
| Outputs | **output name** | | **Datatype** | **Format** |
| fact | | double | N/A |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| identifier and name | RF4- Cosine method | | | |
| Summary | The program must allow to calculate the cosine of a number by means of the simplified definition of the Taylor series and return the result. | | | |
| Inputs | **input name** | **Datatype** | | **Condition valid values** |
| x | double | | Must be the number(radians) to calculate |
| Result or Postcondition | The program calculates the cosine of the entered number and returns the result. | | | |
| Outputs | **output name** | | **Datatype** | **Format** |
| cos | | double | N/A |