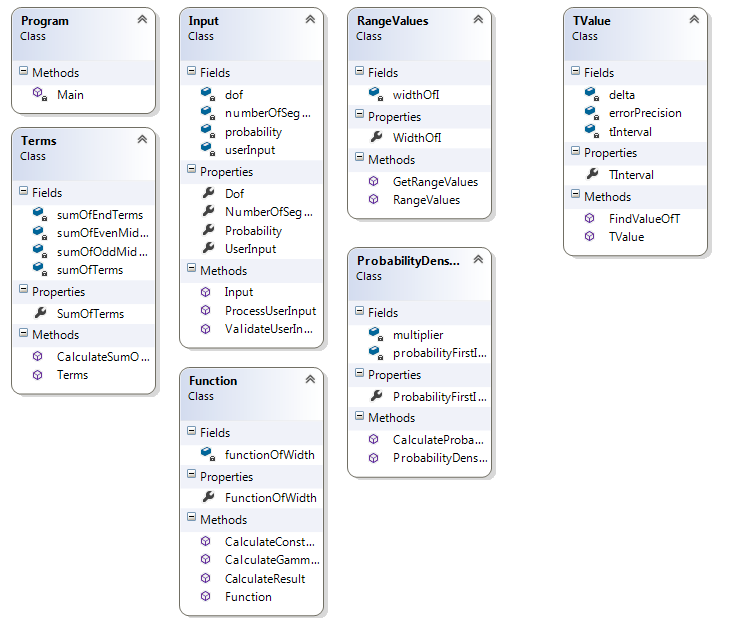
# Operational Specification Template

|  |  |  |  |
| --- | --- | --- | --- |
| **Student** | Hristina Koleva | **Date** | 02.07.2013 |
| **Program** | PSP Assignment 6 | **Program #** | 6 |
| **Instructor** | Valentina Ivanova | **Language** | C# |

|  |  |  |  |
| --- | --- | --- | --- |
| Scenario Number | **1** | **User Objective** | Determine the value for probability density |
| **Scenario Objective** | ***User enters a number between 1 and 3 to select test data for probability and degrees of freedom*** | | |
| **Source** | **Step** | **Action** | **Comments** |
| **User** | 1 | User launches the calculating application |  |
| Application | 2 | Application is launched |  |
| User | 3 | User enters required input value  User presses Enter |  |
| Application | 4 | Application visualizes the result for the input value |  |
|  |  |  |  |
| **Scenario Number** | **2** | **User Objective** | Determine the value for probability density |
| **Scenario Objective** | *Application validates user input* | | |
| **Source** | **Step** | **Action** | **Comments** |
| **User** | 1 | User launches the calculating application |  |
| Application | 2 | Application is launched |  |
| User | 3 | User enters required input value  User presses Enter |  |
| Application | 4 | Application prompt the user that the calculations cannot proceed with the value entered since it is not a valid one |  |
|  |  |  |  |
|  |  |  |  |

# Functional Specification Template



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Student** | Hristina Koleva | **Date** | 02.07.201 |  |  |
| **Program** | PSP Assignment 6 | **Program #** | 6 |  |  |
| **Instructor** | Valentina Ivanova | **Language** | C# |  |  |
|  |  |  | | | |
| Class Name | Program | | | | |
| **Parent Class** |  | | | | |
|  |  | | | | |
|  |  | | | | |
|  |  | | | | |
|  |  |  | | | |
| **Attributes** | | | | | |
|  | **Declaration** | **Description** | | | |
|  |  |  | | | |
|  |  |  | | | |
|  |  |  | | | |
|  |  |  | | | |
|  |  |  | | | |
| **Items** | | | | | |
|  | **Declaration** | **Description** | | | |
|  | Main() | Launches the application and calls the execution of the program | | | |

|  |  |  |
| --- | --- | --- |
| Class Name | UserInput | |
| **Parent Class** |  | |
|  |  | |
|  |  | |
|  |  | |
|  |  |  |
| **Attributes** | | |
|  | **Declaration** | **Description** |
|  | dof | User input to select the degrees of freedom to be used in the calculations of *Student’s*  *t-distribution probability density function* |
|  | probability | Required probability to achieve with tInterval |
|  | NumberOfSegments | Initially = 10 |
|  |  |  |
| **Items** | | |
|  | **Declaration** | **Description** |
|  | ValidateUserInput() | Validates the user input from key board as a numeric value |
|  | ProcessUserInput() | Sets probability depending on the entered *dof* |

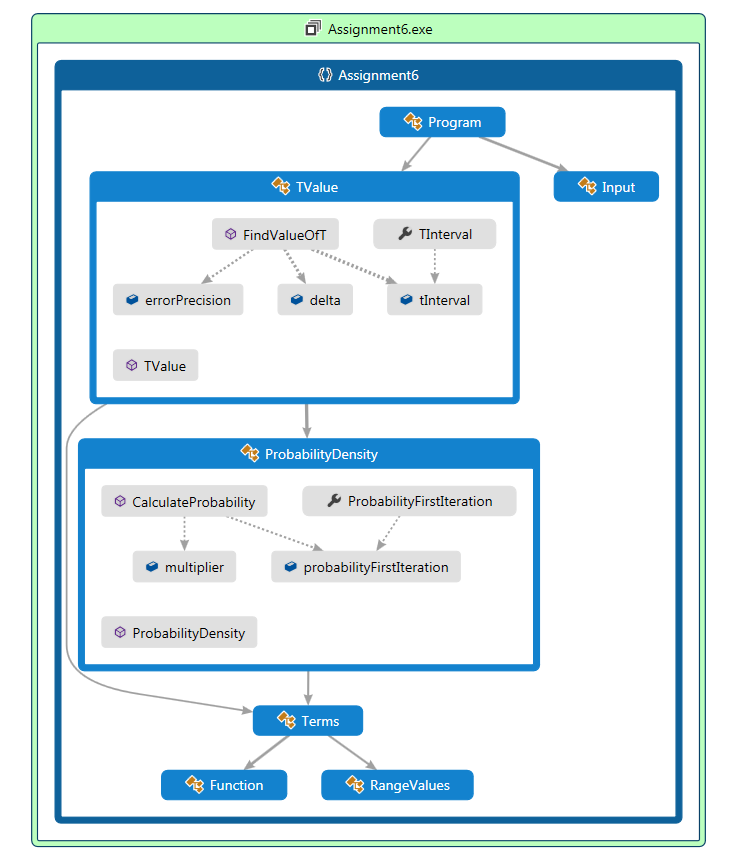
|  |  |  |
| --- | --- | --- |
| Class Name | RangeValues | |
| **Parent Class** |  | |
|  |  | |
|  |  | |
|  |  | |
|  |  |  |
| **Attributes** | | |
|  | **Declaration** | **Description** |
|  | dof | Uses *dof* from *UserInput* |
|  | WidthOfI | The range of function parameters |
|  | tInterval | End of widthOfIrange in the current iteration of finding tInterval |
|  | numberOfSegments |  |
|  |  |  |
| **Items** | | |
|  | **Declaration** | **Description** |
|  | GetRangeValues() | widthOfI[i] = i \* tInterval / numberOfSegments; |

|  |  |  |
| --- | --- | --- |
| Class Name | Terms | |
| **Parent Class** |  | |
|  |  | |
|  |  | |
|  |  | |
|  |  |  |
| **Attributes** | | |
|  | **Declaration** | **Description** |
|  | dof | User input to select the degrees of freedom to be used in the calculations of *Student’s*  *t-distribution probability density function* |
|  | functionOfWidth | The range of function parameters |
|  | tInterval | End of *functionOfWidth* range |
|  | SumOfEndTerms | Calculate F(0) + F(X) |
|  |  |  |
| **Items** | | |
|  | **Declaration** |  |
|  | CalculateSumOfTerms() |  |

|  |  |  |
| --- | --- | --- |
| Class Name | Function | |
| **Parent Class** |  | |
|  |  | |
|  |  | |
|  |  | |
|  |  |  |
| **Attributes** | | |
|  | **Declaration** | **Description** |
|  | dof | User input to select the degrees of freedom to be used in the calculations of *Student’s*  *t-distribution probability density function* |
|  | functionOfWidth | The range of function parameters |
|  | tInterval | End of *functionOfWidth* range |
|  | WidthOfI | X as returned by *RangeValues* |
|  |  |  |
| **Items** | | |
|  | **Declaration** | **Description** |
|  | CalculateGamma() | Calculates the constant function part of gamma function of parameter *dof* |
|  | CalculateConstant() |  |
|  | CalculateResult() | Calculates F(X) for each *widthOfI* as m*ultiplies the result of CalculateConstant() and the variable part of the function* and returns the output of the function |

|  |  |  |
| --- | --- | --- |
| Class Name | ProbabilityFunction | |
| **Parent Class** | Program | |
|  |  | |
|  |  | |
|  |  | |
|  |  |  |
| **Attributes** | | |
|  | **Declaration** | **Description** |
|  | dof | *From UserInput* |
|  | numberOfSegments | *From UserInput* |
|  | Tinterval | *From TValue* |
|  | FunctionOfWidth | Uses result calculated in *CalculateResultFunction()* of *CalculateFunction* class |
|  | probabilityFirstIterationN | The output result of the application, calculated in *CalculateProbability() method* using *firstNumberOfSegments* |
|  | Multiplier | Multiplier = tInterval/numberOfSegments\*3 |
|  |  |  |
| **Items** | | |
|  | **Declaration** | **Description** |
|  | CalculateProbability() | Calculates the output of the application by summarizing the terms and multiplying them by the Width coefficient |

# Logic Specification Template



|  |  |  |  |
| --- | --- | --- | --- |
| Student | Hristina Koleva | **Date** | 02.07.2013 |
| Program | PSP Assignment 6 | **Program #** | 6 |
| Instructor | Valentina Ivanova | **Language** | C# |

|  |  |
| --- | --- |
| **Design** |  |
| **References** |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Parameters** | dof |
|  | tInerval |
|  | functionOfWidth |
|  | errorTolerance |
|  | isMoreAccurate |
|  | numberOfSegments |
|  | probabilityFirstIteration |
|  | probabilityNextIteration |
|  | sumOfTerms |
|  | widthOfI |
|  | isValidUserInput |
|  | userInput |
|  | sumOfOddTerms |
|  | sumOfEvenTerms |
|  | multiplier |
|  |  |

|  |
| --- |
| ***ValidateUserInput****()* |
| Read userInput |
| While userInput is valid |
| Call ***ProcessUserInput()*** |
| Else Return |
|  |
| ***ProcessUserInput****()* |
| Case IserInputof |
| UserInput = 1 : probability = 0.20; dof = 6 |
| UserInput = 2: probability = 0.45; dof = 15 |
| UserInput = 3: pro = 0.495; dof = 4 |
|  |
| ***FindValueOfT()*** |
| Tinterval = 1.0 |
| Delta = 0.5 |
| ErrorPrecision = 0.0000001 |
| Call ***CalculateProbability()*** *return probabilityFirstIteration* |
|  |
| ***While (NOT !(probability – probability <= E))*** |
|  |
| **IF**(Math.Sign(probability – probabilityFirstIteration) <> Math.Sign(errorPrecision) |
| { |
| Delta = delta/2 |
| If (probability > probabilityFirstIteration) |
| { |
| Tinterval = tInterval + delta |
| Call ***CalculateProbability()*** |
| } |
| Else if (probability < probabilityFirstIteration) |
| { |
| Tinterval = tInterval - delta |
| Call ***CalculateProbability()*** |
| } |
| } |
| Else if (Math.Sign(probability – probabilityFirstIteration) == Math.Sign(errorPrecision) |
| { |
| If (probability > probabilityFirstIteration) |
| { |
| Tinterval = tInterval + delta |
| Call ***CalculateProbability()*** |
| } |
| Else if (probability < probabilityFirstIteration) |
| { |
| Tinterval = tInterval - delta |
| Call ***CalculateProbability()*** |
| } |
| } |
| Return; |
| ***GetRangeValues****()* |
| For I = 0; I <numberOfsegments+1; I ++ |
| WidthOfI[i] = i\*tInterval/numberOfSegments |
|  |
| ***CalculateGamma****(dof)* |
| If dof == 1 return 1 |
| If dof == 0.5 return |
| Return (dof-1)\*CalculateGamma(dof+1) |
|  |
| ***CalculateConstant****()* |
| Return CalculateGamma((dof + 1) / 2) /\*CalculateGamma(dof/2) |
|  |
| **CalculateResult**() |
| For I = 0; I < numberOfSegments +1; i++ |
| FunctionOfWidth[i] = CalculateConstant(dof)\* |
|  |
| ***CalculateSumOfTerms****()* |
| SumOfEndTerms = funtionOfWidth[0] + functionOfWidth[tInterval] |
|  |
| For I = 1; I < numberOfSegments; I+=2 |
| sumOfOddTerms[i] = functionOfWidth[i]\*4 |
| SumOfTerms += sumOfOddTerms[i] |
|  |
| For J = 2; J < numberOfSegments; J +=2 |
| sumOfEvenTerms[j] = functionOfWidth[i]\*2; |
|  |
| SumOfTerms += sumOfEndTerms; |
|  |
| ***CalculateProbability()*** |
|  |
| Call ***CalculateSumOfTerms()*** |
|  |
| Multiplier = tInterval/numberOfSegments\*3 |
|  |
| ProbabilityFirstIteration = multiplier\*sumOfTerms |
|  |
| Console.WriteLine(TInterval = {0:F5}, tInterval); |