

Logic Specification Template

Student José Alberto Esquivel Patiño

Program # 4

Class Name IOHandler

Method Name readValue

Parameters sPrompt : String

sErrorMessage : String

patValidStructure : Pattern

declare sValue
declare matValidator
assign patValidStructure.matcher() to matValidator
print sPrompt
read sValue
trim sValue
while sValue does NOT match with matValidator
print sErrorMessage
print sPrompt
read sValue
return sValue

Class Name	AreaUnderTDistribution
Method Name	main
Parameters	

declare and initialise variable areCalculator of type AreaUnderTDistribution
declare and initialise variable ioHandler of type IOHandler
call readValue on ioHandler with parameters: "Introduce el valor límite de integración x: (debe de ser numérico y mayor o igual a 0)", sINVALID_REAL_NUMBER, Pattern.compile("(\\d+(\\.\\d+)?)")
parse the returned value to double
assign the returned value to class variable dX on areCalculator
call readValue on ioHandler with parameters: "Introduce el valor de los grados de libertad dof: (debe de ser numérico entero y mayor a 0)", sINVALID_INTEGER, Pattern.compile("\\d*[1-9]\\d*")
parse the returned value to Integer
assign the returned value to class variable iDof on areCalculator
call the class function calculate on areCalculator
print areCalculator

Class Name	AreaUnderTDistribution
Method Name	gamma
Parameters	dX : double

if dX is 1
return 1
else if dX is 1/2
return square root of PI
else
return (dX - 1) * gamma(dX - 1)

Class Name	AreaUnderTDistribution
Method Name	tStudent
Parameters	dX : double

```
return ( gamma( (iDof + 1) / 2 ) / ( (dof*PI)^(1/2) * gamma( dof / 2 ) ) ) * ( 1 + (dX^2/iDof) ) ^ ( (iDof + 1)/(-2) )
```

Class Name	AreaUnderTDistribution
Method Name	calculate
Parameters	

```
declare iNumSeg and initialise with 8
declare dW and initialise with dX / iNumSeg
declare dE and initialise with 0.0000001
declare dPreviousP
call simpson with parameters : dX, dW, iNumSeg
assign returned value to dP
do the following
    assign dP to dPreviousP
    double the value of iNumSeg
    assign dX / iNumSeg to dW
    call simpson with parameters : dX, dW, iNumSeg
    assign returned value to dP
while the absolute value of ( dP - dPreviousP ) > dE
```

Class Name	AreaUnderTDistribution
Method Name	simpson
Parameters	dX : double dW : double iNumSeg - 1 : Integer
declare dSum4W and initialise with 0	
declare dSum2W and initialise with 0	
declare iCont and initialise with 1	
while iCont is less than or equal iNumSeg - 1	
dSum4W += 4 * tStudent(iCont * dW)	
assign iCont + 2 to iCont	
assign 2 to iCont	
while iCont is less than or equal iNumSeg - 2	
dSum2W += 2 * tStudent(iCont * dW)	
assign iCont + 2 to iCont	
return (dW / 3) * (tStudent(0) + dSum4W + dSum2W + tStudent(dX))	

Class Name	AreaUnderTDistribution
Method Name	toString
declare variable sFormat	
assign “x = %.5f\ndof = %d\np = %.5f” to sFormat	
return sFormat.format(dX, iDof, dP);	