

20190021_20190042_20190224

TEST CASE REPORT

GENERAL INFORMATION			
Test Date:	04/27/2023	Test Case Number:	1
Test Case Description:	Default Constructor initialize quarter and year with values of now		
Results:	Passed Failed		
TEST			
Input Specifications:	Attributes of the year now 2023, quarter now 2		
Procedural Steps:	Using default constructor and test each value with the right value now , test equality of (2023, getYear()), and (2, getQuarter()).		
Expected Results of Case:	Ensure that the Default constructor work well and getYear, getQuarter work as expected		
ACTUAL RESULTS			
Output Specifications and comments :	Test Passed, Output of Year is 2023 as expected, Output of Quarter is 2 as expected.		

GENERAL INFORMATION			
Test Date:	04/27/2023	Test Case Number:	2
Test Case Description:	Constructor with Date object to can get year and quarter from it		
Results:	Passed Failed		
TEST			
Input Specifications:	Attributes of the year now 2023, quarter now 2		
Procedural Steps:	Using default Date object as other constructors deprecated and test each value with the right value now , test equality of (2023, getYear()) and (2, getQuarter()).		
Expected Results of Case:	Ensure that the constructor with date object work well and getYear, getQuarter work as expected		
ACTUAL RESULTS			
Output Specifications and comments :	Test Passed, Output of Year is 2023 as expected, Output of Quarter is 2 as expected.		

GENERAL INFORMATION			
Test Date:	04/27/2023	Test Case Number:	3
Test Case Description:	Constructor with quarter, Year		
Results:	Passed Failed with expected= illegalArgumentException		
TEST			
Input Specifications:	Attributes of the year now 20000, quarter 3		
Procedural Steps:	Using default Year object with out of range 20000 , quarter 3		
Expected Results of Case:	illegalArgumentException that 20000 is out of range (1900,9999)		
ACTUAL RESULTS			
Output Specifications and comments :	Test Passed with expected=illegalArgumentException. Insufficient Documented function that can't tell what will function throw when take input out of range.		

GENERAL INFORMATION			
Test Date:	04/27/2023	Test Case Number:	4
Test Case Description:	Constructor with quarter, year object		
Results:	Passed <div>Failed</div>		
TEST			
Input Specifications:	Attributes of the year =2023, quarter = 5 out of range(1,4)		
Procedural Steps:	Using year object with 2023 and integer quarter = 5 .		
Expected Results of Case:	Expect to take quarter 5 as the maximum number to be 4 but it take 5 as it and store in quarter		
ACTUAL RESULTS			
Output Specifications and comments :	Test Failed, Output of Year is 2023 as expected, Output of Quarter is 5 but this out of range that documented. Insufficient Documentation when pass quarter value out of the range.		

GENERAL INFORMATION			
Test Date:	04/27/2023	Test Case Number:	5
Test Case Description:	Constructor with Date object and TimeZone object		
Results:	Passed Failed		
TEST			
Input Specifications:	Default Date , Default TimeZone		
Procedural Steps:	Using default Date object ,TimeZone object and assert equal to time now 2023 and quarter 2.		
Expected Results of Case:	Year = 2023, quarter=2		
ACTUAL RESULTS			
Output Specifications and comments :	Test Passed, Output of Year is 2023 as expected, Output of Quarter is 2 as expected.		

GENERAL INFORMATION			
Test Date:	04/27/2023	Test Case Number:	6
Test Case Description:	CompareTo		
Results:	Passed Failed		
TEST			
Input Specifications:	Quarter(1,1998) Quarter(4,1999)		
Procedural Steps:	Create two quarters with different times to test negativity that one come before other.		
Expected Results of Case:	-1 that first quarter come before second quarter		
ACTUAL RESULTS			
Output Specifications and comments :	Test Passed, get -1 as expected.		

GENERAL INFORMATION			
Test Date:	04/27/2023	Test Case Number:	7
Test Case Description:	Equals		
Results:	Passed Failed		
TEST			
Input Specifications:	Quarter(4,1900) Quarter(4,1900)		
Procedural Steps:	Passing two quarters object with same Time period to test happy scenario		
Expected Results of Case:	True as expected		
ACTUAL RESULTS			
Output Specifications and comments :	Test Passed, True.		

GENERAL INFORMATION			
Test Date:	04/27/2023	Test Case Number:	8
Test Case Description:	Equals		
Results:	Passed <div>Failed</div>		
TEST			
Input Specifications:	Quarter(1,1900) Calender with same time period		
Procedural Steps:	Create quarter object and calender object with same time period.		
Expected Results of Case:	Should be equal, Assert True		
ACTUAL RESULTS			
Output Specifications and comments :	Test Failed, AssertionError . Insufficient in document what is arbitrary objects to pass to the function		

GENERAL INFORMATION			
Test Date:	04/27/2023	Test Case Number:	9
Test Case Description:	GetFirstMilliSecond		
Results:	Passed Failed		
TEST			
Input Specifications:	Quarter object Calender object		
Procedural Steps:	Set time zone in calender, calculate actual First millisecond, expected first millisecond.		
Expected Results of Case:	Equals as expected		
ACTUAL RESULTS			
Output Specifications and comments :	Test Passed, True as expected.		

GENERAL INFORMATION			
Test Date:	04/27/2023	Test Case Number:	10
Test Case Description:	GetLastMilliSecond		
Results:	Passed Failed		
TEST			
Input Specifications:	Quarter object Calender object		
Procedural Steps:	Set time zone in calender, calculate actual Last millisecond, expected Last millisecond.		
Expected Results of Case:	Equals as expected		
ACTUAL RESULTS			
Output Specifications and comments :	Test Passed, True as expected.		

GENERAL INFORMATION			
Test Date:	04/27/2023	Test Case Number:	11
Test Case Description:	parseQuarter		
Results:	Passed Failed		
TEST			
Input Specifications:	“2019-Q3” , “2019 Q3” , “2019/Q3” , “2019,Q3” “Q3-2019” , “Q3 2019” , “Q3/2019” , “Q3,2019”		
Procedural Steps:	Test each string format to parse.		
Expected Results of Case:	Get value of quarter , value of year. Get values as expected		
ACTUAL RESULTS			
Output Specifications and comments :	Test Passed, quarter=3, year=2019.		

GENERAL INFORMATION			
Test Date:	04/27/2023	Test Case Number:	12
Test Case Description:	parseQuarter		
Results:	Passed <div>Failed</div>		
TEST			
Input Specifications:	“2019_Q3” , “Q3_2019” , “Q3” , “2019” , “Q 2019”		
Procedural Steps:	Test different format of strings		
Expected Results of Case:	Get Exceptions not documented		
ACTUAL RESULTS			
Output Specifications and comments :	NumberFormatException, TimePeriodFormat Exception Insufficient Info in the documentation.		

GENERAL INFORMATION				
Test Date:	04/27/2023		Test Case Number:	13
Test Case Description:	getQuarter			
Results:	Passed Failed			
TEST				
Input Specifications:	1 , 2 , 3 , 4			
Procedural Steps:	Quarter with any integer within the range.			
Expected Results of Case:	Work as expected and get the right value.			
ACTUAL RESULTS				
Output Specifications and comments :	Passed, True as expected.			

GENERAL INFORMATION			
Test Date:	04/27/2023	Test Case Number:	14
Test Case Description:	getQuarter		
Results:	Passed <div>Failed</div>		
TEST			
Input Specifications:	-1 , 7		
Procedural Steps:	Quarter with any integer out of the range.		
Expected Results of Case:	Can't handle this case and take value as it. Expect to take -1 as 1 , 7 as 4 but that not actual do.		
ACTUAL RESULTS			
Output Specifications and comments :	Failed, Can't handle integers out of the range.		

GENERAL INFORMATION			
Test Date:	04/27/2023	Test Case Number:	15
Test Case Description:	getYear		
Results:	Passed Failed		
TEST			
Input Specifications:	2027		
Procedural Steps:	Create quarter object with year 2027		
Expected Results of Case:	2027 as expected.		
ACTUAL RESULTS			
Output Specifications and comments :	Passed, True as expected. Works well with the values within the range.		

GENERAL INFORMATION			
Test Date:	04/27/2023	Test Case Number:	16
Test Case Description:	getYear		
Results:	Passed Failed expected=illegalArgumentException		
TEST			
Input Specifications:	1000, 20000		
Procedural Steps:	Create quarter object with year 1000 out of the range		
Expected Results of Case:	Expect to take 1000 as a min value 1900, 20000 as max value 9999 or document how to handle this case. but it throws exception not documented.		
ACTUAL RESULTS			
Output Specifications and comments :	Passed, with expected=illegalArgumentException. Insufficient Info in document that throw exception.		

GENERAL INFORMATION			
Test Date:	04/27/2023	Test Case Number:	17
Test Case Description:	previous		
Results:	Passed Failed		
TEST			
Input Specifications:	Quarter(1, 1900)		
Procedural Steps:	Create quarter object with quarter=1, year=1900		
Expected Results of Case:	Null.		
ACTUAL RESULTS			
Output Specifications and comments :	Passed, Null as expected.		

GENERAL INFORMATION			
Test Date:	04/27/2023	Test Case Number:	18
Test Case Description:	previous		
Results:	Passed <div>Failed</div>		
TEST			
Input Specifications:	Quarter(7, 2021)		
Procedural Steps:	Create quarter object with quarter=7, year=2021		
Expected Results of Case:	Q3/2021 or document how to handle this case.		
ACTUAL RESULTS			
Output Specifications and comments :	Failed . Output: <Q6/2021>		

GENERAL INFORMATION			
Test Date:	04/27/2023	Test Case Number:	19
Test Case Description:	next		
Results:	Passed Failed		
TEST			
Input Specifications:	Quarter(2, 2023) as default		
Procedural Steps:	Create quarter object with default constructor		
Expected Results of Case:	(3, 2023).		
ACTUAL RESULTS			
Output Specifications and comments :	Passed, True as expected.		

GENERAL INFORMATION			
Test Date:	04/27/2023	Test Case Number:	20
Test Case Description:	next		
Results:	Passed <div>Failed</div>		
TEST			
Input Specifications:	Quarter(-3, 2027)		
Procedural Steps:	Create quarter object with parameterized constructor quarter=-3, year=2027		
Expected Results of Case:	(2, 2027). or document how to handle this case.		
ACTUAL RESULTS			
Output Specifications and comments :	Failed . Output: <Q-2/2027>		

GENERAL INFORMATION			
Test Date:	04/27/2023	Test Case Number:	21
Test Case Description:	getSerialIndex		
Results:	Passed Failed		
TEST			
Input Specifications:	Quarter() as default		
Procedural Steps:	Create quarter object with default constructor, calculate actual serial index by number of passed quarters with equation : Year*4 + quarter		
Expected Results of Case:	True (actualSerialIndex==getSerialIndex).		
ACTUAL RESULTS			
Output Specifications and comments :	Passed, True as expected.		

GENERAL INFORMATION			
Test Date:	04/27/2023	Test Case Number:	22
Test Case Description:	hashCode		
Results:	Passed Failed		
TEST			
Input Specifications:	Quarter() as default , Quarter(Date)		
Procedural Steps:	Create quarter object with default constructor, quarter object with date default		
Expected Results of Case:	True (q1.hashCode == q2.hashCode).		
ACTUAL RESULTS			
Output Specifications and comments :	Passed, True as expected.		

GENERAL INFORMATION			
Test Date:	04/27/2023	Test Case Number:	23
Test Case Description:	toSring		
Results:	Passed Failed		
TEST			
Input Specifications:	Quarter(4, 2050)		
Procedural Steps:	Create quarter object with quarter=4, year=2050		
Expected Results of Case:	“Q4/2050”.		
ACTUAL RESULTS			
Output Specifications and comments :	Passed, True as expected.		

Bugs Report:

Constructor Quarter (int year, int quarter) :

Quarter range (1, 4) : can't detect if user pass -1 or 7 or any number out of the range but work normally with the number passed even if it's a negative number.

Year range (1900,9999) : throw IllegalArgumentException when passing any number out of the range and not documented how to handle this case.

getQuarter :

Get wrongs value with values out of the range (1,4) due to constructor deal with quarter values out of the range.

getYear :

Not Handle Exceptions or documented it if the value is out of the range (1900,9999).

next :

when passing quarter(-3,2027) get <Q-2/2027> must handle and document this case.

previous :

when passing quarter(7,2021) get <Q6/2021> must handle and document this case.

The image displays two screenshots of the Eclipse IDE interface, showing the results of a JUnit test run for the JfreeChart project.

Left Screenshot: The IDE window is titled "eclipse-workspace - JFreeChart/src/org/jfree/data/test/QuarterClassTest.java". The Package Explorer on the left shows the test class hierarchy. The JUnit tab displays the test results, indicating "Runs: 57/57", "Errors: 8", and "Failures: 6". The test cases listed include:

- testGetYear1 (0.001 s)
- testGetYear2 (0.003 s)
- testQuarterCtor10 (0.013 s)
- testQuarterCtor11 (0.000 s)
- testCompareTO2 (0.000 s)
- testCompareTO3 (0.000 s)
- testCompareTO4 (0.000 s)
- testCompareTO5 (0.000 s)
- testCompareTO6 (0.001 s)
- testNext1 (0.000 s)
- testNext2 (0.000 s)
- testNext (0.000 s)
- testToString (0.001 s)
- testParseQuarter10 (0.000 s)
- testParseQuarter11 (0.001 s)
- testParseQuarter12 (0.001 s)
- testCompareTO (0.000 s)
- testPrevious (0.000 s)
- testQuarterDefaultCtor (0.000 s)

Right Screenshot: The IDE window is titled "eclipse-workspace - JFreeChart/src/org/jfree/data/test/QuarterClassTest.java - Eclipse IDE". The Package Explorer on the left shows the test class hierarchy. The JUnit tab displays the test results, indicating "Runs: 57/57", "Errors: 8", and "Failures: 6". The test cases listed include:

- testQuarterDefaultCtor (0.000 s)
- testGetYear (0.000 s)
- testParseQuarter1 (0.000 s)
- testParseQuarter2 (0.000 s)
- testParseQuarter3 (0.000 s)
- testParseQuarter4 (0.000 s)
- testParseQuarter5 (0.000 s)
- testParseQuarter6 (0.000 s)
- testParseQuarter7 (0.000 s)
- testParseQuarter8 (0.000 s)
- testParseQuarter9 (0.001 s)
- testPrevious1 (0.000 s)
- testPrevious2 (0.000 s)
- testPrevious3 (0.001 s)
- testQuarterCtor1 (0.000 s)
- testQuarterCtor2 (0.000 s)
- testQuarterCtor3 (0.000 s)
- testQuarterCtor4 (0.000 s)
- testQuarterCtor5 (0.001 s)
- testQuarterCtor6 (0.000 s)

