**Bug Report:**

|  |  |
| --- | --- |
| Bug Name (Title) | Quarter range |
| **Prioritization** |  |
| Bug Severity | minor |
| Bug Priority | low |
| **Details** |  |
| Steps to Reproduce | Theres more than one way to produce a Quarter  One of them :  Step1 ->initialize two integer values , first one for quarter and second one for year  Step2- > make an obj from Quarter with passing the values of quarter and year |
| Expected Result | The expected result to accept any values for year between (1900 to 9999)  And for quarter between (1 to 4) |
| Actual Result | After trying to make a Quarter obj with different values ,the range of year works right otherwise the range of quarter doesn’t work as we expect it accept any value and doesn’t refuse the values ourt of (1 to 4) range |

|  |  |
| --- | --- |
| method | **Quarter** |
| description | A constractur with 5 different implementatioons |
| test cases | **Quarter**()  **Quarter**(int quarter, int year)  Th expected value to set quarter and year value with range year - the year (1900 to 9999).  quarter - the quarter (1 to 4).  **Quarter**(int quarter,[Year](file:///C:\Users\Omar\Downloads\Assignment%201%20-%202023%20(1)\ModifiedJavadoc\org\jfree\data\time\Year.html) year)  Th expected value to set quarter and year value with range year - the year (1900 to 9999).  quarter - the quarter (1 to 4).  **Quarter**(java.util.Date time)  Constructs a new Quarter, based on a date/time and the default time zone.  **Quarter**(java.util.Date time,  java.util.TimeZone zone)  Constructs a Quarter, based on a date/time and time zone. |
| expected | The expected value to set quarter and year within the range of (1 to 4) for the quarter and (1900 to 9999) for the year |
| actual | The actual value sets quarter with no care for the range but in the range of year |
| status | Passed when you give the fun the right vlue for the year (between its range) |
| additional notes | Thers no |
|  |  |

|  |  |
| --- | --- |
| method | getQuarter( ) |
| description | A function that Returns the quarter |
| test cases | Checking if the given quarter has been setted right or not |
| expected | The expected value to get the quarter value rightly we use 2 for quarter and 2002 for the year |
| actual | The actual value was the same as the expected |
| status | Passed |
| additional notes | Thers no |
|  |  |

|  |  |
| --- | --- |
| method | getYear( ) |
| description | A function that Returns the year |
| test cases | Checking if the given year has been setted right or not |
| expected | The expected value to get the quarter value rightly we use 2 for quarter and 2002 for the year |
| actual | The actual value was the same as the expected |
| status | Passed |
| additional notes | Thers no |
|  |  |

|  |  |
| --- | --- |
| method | Previous( ) |
| description | returns the quarter preceding this on |
| test cases | Take two obj of quarter (the new one and the previous one )and check if previous return the previous value right or not |
| expected | to return the value of the obj declared before the last one |
| actual | Its give as the value of the previous obj |
| status | Passed |
| additional notes | Thers no |
|  |  |

|  |  |
| --- | --- |
| method | next( ) |
| description | Returns the quarter following this one. |
| test cases | Take two obj of quarter (the new one and the next one )and check if previous return the following value right or not |
| expected | to return the value of the obj declared after the one |
| actual | Its give as the value of the following obj |
| status | Passed |
| additional notes | Thers no |
|  |  |
| method | getSerialIndex( ) |
| description | Returns a serial index number for the quarter |
| test cases | We check what the function return with the number of years multiplied by 4 and added to this value the value of quarter |
| expected | To give us the seral index of the (obj(2010)\*4)+2 |
| actual | (2010\*4)+2 |
| status | Passed |
| additional notes | Thers no |
|  |  |

|  |  |
| --- | --- |
| method | Equals( ) |
| description | Tests the equality of this Quarter object to an arbitrary object. Returns true if the target is a Quarter instance representing the same quarter as this object. In all other cases, returns false |
| test cases | Make an obj by arrange fun and another one both with same values and check the equal fun by use it by obj from them and pass another obj to it |
| expected | To return true |
| actual | true |
| status | Passed |
| additional notes | Thers no |
|  |  |

|  |  |
| --- | --- |
| method | hashCode( ) |
| description | Returns a hash code for this object instance |
| test cases | Returns the hash code of an obj and the hash code be the same of the same values |
| expected | Same hash code for the two objects |
| actual | Returns the same hash code or the two objects with the same values |
| status | Passed |
|  |  |

|  |  |
| --- | --- |
| method | compareTo( ) |
| description | Returns an integer indicating the order of this Quarter object relative to the specified object: negative == before, zero == same, positive == after |
| test cases | We use it with 3 cases with first number is bigger in the order ,same order and less than the second one |
| expected | In the first one to return negative which means before  In the second one to return zero which means same  In the third one to return positve which means after |
| actual | Negative  Zero  positive |
| status | passed |
|  |  |

|  |  |
| --- | --- |
| method | toString( ) |
| description | Returns a string representing the quarter |
| test cases |  |
| expected | "Q1/2002” |
| actual | Q1/2002 |
| Status | passed |
|  |  |

|  |  |
| --- | --- |
| method | getFirstMillisecond() |
| description | Returns the first millisecond in the Quarter, evaluated using the supplied calendar (which determines the time zone |
| test cases |  |
| expected | The first millisecond in the Quarte |
| actual | The first millisecond in the Quarte |
| Status | passed |
|  |  |

|  |  |
| --- | --- |
| method | getLastMillisecond( ) |
| description | Returns the last millisecond of the Quarter, evaluated using the supplied calendar (which determines the time zone). |
| test cases |  |
| expected | The last millisecond in the Quarte |
| actual | The last millisecond in the Quarte |
| Status | passed |
|  |  |

|  |  |
| --- | --- |
| method | parseQuarter() |
| description | Parses the string argument as a quarter.  This method should accept the following formats: "YYYY-QN" and "QN-YYYY", where the "-" can be a space, a forward-slash (/), comma or a dash (-). |
| test cases |  |
| expected | YYYY-QN , QN-YYYY |
| actual | the quarter with right value |
| Status | passed |
|  |  |