Name: Omar Waleed Zenhom ID: 20206130

Name: Mohamed Alaa El-Din ID: 20206068

Bug Report

Title: Default constructor

- Case: testQuarterDefaultCtor
  - Expected Parameters: -----------------
  - Sending Parameters:
  - Description: Constructs a new Quarter, based on the current system date/time.
  - Expected Result: Year = 2023 / Quarter = 2
  - Actual Result: Year = 2023 / Quarter = 2
  - Severity: -------------
  - Priority: --------------
  - Status: Passed
  - Additional notes: ----------

Title: Quarter(int quarter, int year) constructor

- Case 1: testQuarterDefaultCtor
  - Expected Parameters: year - the year (1900 to 9999). ----- quarter - the quarter (1 to 4).
  - Sending Parameters: Year = 2002 / Quarter = 2
  - Description: Constructs a new Quarter, set the Quarter/Year.
  - Expected Result: Year = 2002 / Quarter = 2
  - Actual Result: Year = 2002 / Quarter = 2
  - Severity: -------------
  - Priority: --------------
  - Status: Passed
  - Additional notes: ----------
- Case 2: testQuarterIntIntConstructorInvalidQuarter()
  - Expected Parameters: year - the year (1900 to 9999). ----- quarter - the quarter (1 to 4).
  - Sending Parameters: Year = 2023 / Quarter = -5
  - Description: Constructs a new Quarter, set the Quarter/ Year.
  - Expected Result: Year = 2023 / Quarter = “Quarter outside valid range.”
  - Actual Result: Year = 2023 / Quarter = -5
  - Severity: Major
  - Priority: High
  - Status: Failed
  - Additional notes: ----------
- Case 3: testQuarterIntIntConstructorInvalidYear
  - Expected Parameters: year - the year (1900 to 9999). ----- quarter - the quarter (1 to 4).
  - Sending Parameters: Year = 1800 / Quarter = 1
  - Description: Constructs a new Quarter, set the Quarter/ Year.
  - Expected Result: Year = Year did not set. - “Year outside valid range.” / Quarter = 1
  - Actual Result: Year = Year did not set. - “Year outside valid range.” / Quarter = 1
  - Severity: ------------
  - Priority: -------------
  - Status: Passed

    - Additional notes: ----------

Title: Quarter (int quarter, Year year) constructor

- Case 1: testQuarterIntYear
    - Expected Parameters: quarter - the quarter (1 to 4) --------- year - the year (1900 to 9999).
    - Sending Parameters: Year = Year.of(2023) / Quarter = 2
    - Description: Constructs a new Quarter, set the Quarter/ Year from object of Year Class.
    - Expected Result: Year = 2023 / Quarter = 2
    - Actual Result: Year = 2023 / Quarter = 2
    - Severity: -------------
    - Priority: --------------
    - Status: Passed
    - Additional notes: ----------
- Case 2: testQuarterIntYearWithInvalidQuarter
    - Expected Parameters: quarter - the quarter (1 to 4) --------- year - the year (1900 to 9999).
    - Sending Parameters: Year = Year.of(2023) / Quarter = 0
    - Description: Constructs a new Quarter, set the Quarter/ Year from object of Year Class.
    - Expected Result: Year = 2023 / Quarter = "Quarter outside valid range."
    - Actual Result: Year = 2023 / Quarter = 0
    - Severity: Major
    - Priority: Medium
    - Status: Failed
    - Additional notes: ----------
- Case 3: testQuarterIntYearWithInvalidYear
    - Expected Parameters: quarter - the quarter (1 to 4) --------- year - the year (1900 to 9999).
    - Sending Parameters: Year = Year.of(1800) / Quarter = 2
    - Description: Constructs a new Quarter, set the Quarter/ Year from object of Year Class.
    - Expected Result: Year = Year did not set. - "Year outside valid range." / Quarter = 2
    - Actual Result: Year = "Year outside valid range." / Quarter = 2
    - Severity: -------------
    - Priority: --------------
    - Status: Passed
    - Additional notes: ----------

Title: Quarter (java.util.Date time) constructor

- Case 1: testQuarterConstructorWithDate
    - Expected Parameters: time - the date/time
    - Sending Parameters: Date of today based on system
    - Description: Constructs a new Quarter, based on a date/time and the default time zone.
    - Expected Result: Year = 2023 / Quarter = 2
    - Actual Result: Year = 2023 / Quarter = 2
    - Severity: -------------
    - Priority: --------------
    - Status: Passed
    - Additional notes: ----------

Title: Quarter (java.util.Date time, java.util.TimeZone zone) constructor

- Case 1: testDateAndTimeZoneConstructor()

    - Expected Parameters: time - the date/time
    - Sending Parameters: Date of today based on system and the time zone
    - Description: Constructs a Quarter, based on a date/time and time zone.
    - Expected Result: Year = 2023 / Quarter = 2
    - Actual Result: Year = 2023 / Quarter = 2
    - Severity: -------------
    - Priority: --------------
    - Status: Passed
    - Additional notes: ----------

Methods Bug Report:

Title: getQuarter Method

- Case 1: testGetQuarter()
    - Expected Parameters: -----
    - Sending Parameters:
    - Description: Returns the quarter.
    - Expected Result: Quarter = 2
    - Actual Result: Quarter = 2
    - Severity: -------------
    - Priority: --------------
    - Status: Passed
    - Additional notes: ----------

Title: getYear() Method

- Case 1: testgetYear()
    - Expected Parameters: -----
    - Sending Parameters:
    - Description: Returns the Year.
    - Expected Result: Year = 2023
    - Actual Result: Year = 2023
    - Severity: -------------
    - Priority: --------------
    - Status: Passed
    - Additional notes: ----------

Title: previous() Method

- Case 1: testPrevious()
    - Expected Parameters: -----
    - Sending Parameters:
    - Description: Returns the quarter preceding this one.
    - Expected Result: Quarter = 1
    - Actual Result: Quarter = 1
    - Severity: -------------
    - Priority: --------------
    - Status: Passed
    - Additional notes: ----------
- Case 2: testPreviousQ1_1900()

    - Expected Parameters: -----
    - Sending Parameters:
    - Description: The quarter preceding this one (or null if this is Q1 1900).
    - Expected Result: Quarter = 1
    - Actual Result: Null
    - Severity: -------------
    - Priority: --------------
    - Status: Passed
    - Additional notes: ----------
- Case 3: testPreviouswithInvalidQuarter()
    - Expected Parameters: -----
    - Sending Parameters:
    - Description: The quarter preceding this one (or null if this is Q1 1900).
    - Expected Result: Quarter = "Quarter outside valid range."
    - Actual Result: Quarter = 4
    - Severity: Major
    - Priority: High
    - Status: Failed
    - Additional notes: ----------

Title: next() Method

- Case 1: testnext()
    - Expected Parameters: -----
    - Sending Parameters:
    - Description: Returns the quarter following this one.
    - Expected Result: Quarter = 3
    - Actual Result: Quarter = 3
    - Severity: -------------
    - Priority: --------------
    - Status: Passed
    - Additional notes: ----------
- Case 2: testnextQ4_9999()
    - Expected Parameters: -----
    - Sending Parameters:
    - Description: The quarter preceding this one (or null if this is Q4 9999).
    - Expected Result: Quarter = 1
    - Actual Result: Null
    - Severity: -------------
    - Priority: --------------
    - Status: Passed
    - Additional notes: ----------
- Case 3: testnextwithInvalidQuarter()
    - Expected Parameters: -----
    - Sending Parameters:
    - Description: The quarter preceding this one (or null if this is Q4 9999).
    - Expected Result: Quarter = "Quarter outside valid range."
    - Actual Result: Quarter = 1
    - Severity: Major
    - Priority: High

    - Status: Failed
    - Additional notes: ----------

Title: getSerialIndex() Method

- Case 1: testGetSerialIndex()
    - Expected Parameters: -----
    - Sending Parameters:
    - Description: Returns a serial index number for the quarter.
    - Expected Result: true
    - Actual Result: true
    - Severity: -------------
    - Priority: --------------
    - Status: Passed
    - Additional notes: ----------

Title: getSerialIndex() Method

- Case 1: testGetSerialIndex()
    - Expected Parameters: -----
    - Sending Parameters:
    - 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.
    - Expected Result: true
    - Actual Result: true
    - Severity: -------------
    - Priority: --------------
    - Status: Passed
    - Additional notes: ----------

Title: "testEquals" Method

- Case 1: testEquals

    - ⍰ Expected Parameters: Quarter objects with same quarter and year
    - ⍰ Sending Parameters: Quarter objects with same quarter and year
    - ⍰ Description: The test checks if the equals() method returns true when comparing two Quarter objects with the same quarter and year.
    - ⍰ Expected Result: The equals() method should return true
    - ⍰ Actual Result: The equals() method returned true
    - ⍰ Severity: Low
    - ⍰ Priority: Medium
    - ⍰ Status: Open
    - ⍰ Additional notes: No further action is required for this case.

- Case 2: testEquals

    - Expected Parameters: Quarter objects with different quarters but same year

  - Sending Parameters: Quarter objects with different quarters but same year
  - Description: The test checks if the equals() method returns false when comparing two Quarter objects with different quarters but the same year.
  - Expected Result: The equals() method should return false
  - Actual Result: The equals() method returned false
  - Severity: Low
  - Priority: Medium
  - Status: Open
  - Additional notes: No further action is required for this case.

- Case 3: testEquals

  - ⍰ Expected Parameters: Quarter objects with same quarter but different years
  - ⍰ Sending Parameters: Quarter objects with same quarter but different years
  - ⍰ Description: The test checks if the equals() method returns false when comparing two Quarter objects with the same quarter but different years.
  - ⍰ Expected Result: The equals() method should return false
  - ⍰ Actual Result: The equals() method returned false
  - ⍰ Severity: Low
  - ⍰ Priority: Medium
  - ⍰ Status: Open
  - ⍰ Additional notes: No further action is required for this case.

- Case 4: testEquals

  - ⍰ Expected Parameters: Quarter object and object of another class
  - ⍰ Sending Parameters: Quarter object and a string object
  - ⍰ Description: The test checks if the equals() method returns false when comparing a Quarter object with an object of another class.
  - ⍰ Expected Result: The equals() method should return false
  - ⍰ Actual Result: The equals() method returned false
  - ⍰ Severity: Low
  - ⍰ Priority: Medium
  - ⍰ Status: Open
  - ⍰ Additional notes: No further action is required for this case.

Title: testHashCode() Method

- Case 1: Same object, same hash code

  - Expected Parameters: Two Quarter objects with the same quarter and year.
  - Sending Parameters: Two Quarter objects (quarter1, quarter2) created with the same quarter and year.
  - Description: Testing if the hash code of the same object is always the same.
  - Expected Result: The hash code of quarter1 and quarter2 should be the same.
  - Actual Result: The hash code of quarter1 and quarter2 is the same.
  - Severity: Low
  - Priority: Medium
  - Status: Passed
  - Additional notes: None

- Case 2: Different object but same quarter and year, same hash code

- Expected Parameters: Two Quarter objects with the same quarter and year.
- Sending Parameters: Two Quarter objects (quarter1, quarter2) created with the same quarter and year.
- Description: Testing if the hash code of different objects with the same quarter and year is always the same.
- Expected Result: The hash code of quarter1 and quarter2 should be the same.
- Actual Result: The hash code of quarter1 and quarter2 is the same.
- Severity: Low
- Priority: Medium
- Status: Passed
- Additional notes: None

- Case 3: Different object and different quarter, different hash code
    - Expected Parameters: Two Quarter objects with different quarters and years.
    - Sending Parameters: Two Quarter objects (quarter1, quarter3) created with different quarters and the same year.
    - Description: Testing if the hash code of different objects with different quarters is always different.
    - Expected Result: The hash code of quarter1 and quarter3 should be different.
    - Actual Result: The hash code of quarter1 and quarter3 is different.
    - Severity: Low
    - Priority: Medium
    - Status: Passed
    - Additional notes: None

- Case 4: Different object and different year, different hash code
    - Expected Parameters: Two Quarter objects with different years and quarters.
    - Sending Parameters: Two Quarter objects (quarter1, quarter4) created with different years and the same quarter.
    - Description: Testing if the hash code of different objects with different years is always different.
    - Expected Result: The hash code of quarter1 and quarter4 should be different.
    - Actual Result: The hash code of quarter1 and quarter4 is different.
    - Severity: Low
    - Priority: Medium
    - Status: Passed
    - Additional notes: None

Title: "testCompareTo() Method"

- Case 1: testCompareTo_Q1BeforeQ2
    - Expected Parameters: q1 = Quarter(2, 2021), q2 = Quarter(3, 2021)
    - Sending Parameters: q1, q2
    - Description: Test if the result of q1.compareTo(q2) is -1, which means q1 is before q2.
    - Expected Result: -1
    - Actual Result: -1
    - Severity: Minor
    - Priority: Low
    - Status: Passed
    - Additional notes:

- Case 2: testCompareTo_Q2AfterQ1
  - Expected Parameters: q1 = Quarter(2, 2021), q2 = Quarter(3, 2021)
  - Sending Parameters: q2, q1
  - Description: Test if the result of q2.compareTo(q1) is 1, which means q2 is after q1.
  - Expected Result: 1
  - Actual Result: 1
  - Severity: Minor
  - Priority: Low
  - Status: Passed
  - Additional notes:
- Case 3: testCompareTo_Q1BeforeQ4
  - Expected Parameters: q1 = Quarter(2, 2021), q4 = Quarter(1, 2021)
  - Sending Parameters: q1, q4
  - Description: Test if the result of q1.compareTo(q4) is 1, which means q4 is before q1 in the same year.
  - Expected Result: 1
  - Actual Result: 1
  - Severity: Minor
  - Priority: Low
  - Status: Passed
  - Additional notes:
- Case 4: testCompareTo_Q1BeforeQ3
  - Expected Parameters: q1 = Quarter(2, 2021), q3 = Quarter(2, 2022)
  - Sending Parameters: q1, q3
  - Description: Test if the result of q1.compareTo(q3) is -1, which means q1 is before q3 even though the years are different.
  - Expected Result: -1
  - Actual Result: -1
  - Severity: Minor
  - Priority: Low
  - Status: Passed
  - Additional notes:
- Case 5: testCompareTo_Q1SameAsQ1
  - Expected Parameters: q1 = Quarter(2, 2021)
  - Sending Parameters: q1, q1
  - Description: Test if the result of q1.compareTo(q1) is 0, which means q1 is the same as q1.
  - Expected Result: 0
  - Actual Result: 0
  - Severity: Minor
  - Priority: Low
  - Status: Passed
  - Additional notes:

Title: "testToString" Method

- Case 1: Missing/invalid quarter parameter

- Expected Parameters: quarter with a value between 1-4 and a valid year

- Sending Parameters: quarter value of 0 and year of 2021

- Description: The test is checking if the method can handle an invalid quarter value, which should not be possible since quarters should be between 1-4.

- Expected Result: "Quarter not valid"

- Actual Result: "Q0/2021"

- Severity: High

- Priority: High

- Status: Open

- Additional notes: The method is not properly validating the input and allowing invalid quarter values to be used in creating a Quarter object. A fix is needed to properly validate the input and handle invalid values.

Title: "testGetFirstMillisecond" Method

- Case 1: "testGetFirstMillisecond" method with valid input

  - Expected Parameters: Quarter object and Calendar object with valid year, month and date.
  - Sending Parameters: Quarter q = new Quarter(3, 2021); Calendar cal = Calendar.getInstance();
  - Description: Test whether the first millisecond of the quarter is calculated correctly for a valid input.
  - Expected Result: The first millisecond of the quarter should match the calendar's time in milliseconds.
  - Actual Result: The test passed successfully.
  - Severity: Low
  - Priority: Medium
  - Status: Passed
  - Additional notes: The input is valid, and the method is working correctly.

- Case 2: "testGetFirstMillisecond" method with invalid Quarter object

  - Expected Parameters: Quarter object with invalid quarter or year.
  - Sending Parameters: Quarter q = new Quarter(-1, 2021);
  - Description: Test whether the method handles an invalid Quarter object correctly.
  - Expected Result: The method should throw an IllegalArgumentException.
  - Actual Result: The test failed. The method did not throw an exception.
  - Severity: Medium
  - Priority: High
  - Status: Failed
  - Additional notes: The method should handle invalid inputs and throw an exception to indicate the error.

- Case 3: "testGetFirstMillisecond" method with invalid Calendar object

  - Expected Parameters: Quarter object and Calendar object with invalid year, month, or date.
  - Sending Parameters: Calendar cal = Calendar.getInstance(); cal.set(Calendar.MONTH, 13);
  - Description: Test whether the method handles an invalid Calendar object correctly.
  - Expected Result: The method should throw an IllegalArgumentException.

- Actual Result: The test failed. The method did not throw an exception.
- Severity: Medium
- Priority: High
- Status: Failed
- Additional notes: The method should handle invalid inputs and throw an exception to indicate the error.

Title: Bug report for "testGetLastMillisecond" method

- Case 1: testGetLastMillisecond

    - Expected Parameters: Quarter object and Calendar object.
    - Sending Parameters: Quarter object with quarter 3, year 2021, Calendar object with year 2022, month July, and day 1.
    - Description: This test case verifies whether the getLastMillisecond method returns the last millisecond of the current quarter, given a calendar object.
    - Expected Result: The expected result is that the last millisecond returned by the method is the same as the last millisecond of the quarter.
    - Actual Result: The method returns the last millisecond of the next year's July month instead of the last millisecond of the current quarter.
    - Severity: Major
    - Priority: High
    - Status: Open
    - Additional notes: The method is not returning the expected result, indicating a bug in the implementation. The test case needs to be modified to create the quarter object with the correct quarter and year values, and the method needs to be fixed to return the last millisecond of the current quarter.

Title: Bug report for testParseQuarter() method

Case 1: testParseQuarter()

- Expected Parameters: Valid input string representing a quarter, e.g., "2023-Q3"
- Sending Parameters: input1 = "2023-Q3"
- Description: Testing the parsing of a valid input string into a Quarter object.
- Expected Result: The Quarter object with a quarter of 3 and a year of 2023.
- Actual Result: The Quarter object with a quarter of 3 and a year of 2023 was returned.
- Severity: Low
- Priority: Medium
- Status: Passed
- Additional notes:

Case 2: testParseQuarter()

- Expected Parameters: Valid input string representing a quarter, e.g., "Q2 2022"
- Sending Parameters: input2 = "Q2 2022"
- Description: Testing the parsing of a valid input string into a Quarter object.
- Expected Result: The Quarter object with a quarter of 2 and a year of 2022.
- Actual Result: The Quarter object with a quarter of 2 and a year of 2022 was returned.
- Severity: Low
- Priority: Medium
- Status: Passed

- Additional notes:

Case 3: testParseQuarter()

- Expected Parameters: Valid input string representing a quarter, e.g., "Q1/2000"
- Sending Parameters: input3 = "Q1/2000"
- Description: Testing the parsing of a valid input string into a Quarter object.
- Expected Result: The Quarter object with a quarter of 1 and a year of 2000.
- Actual Result: The Quarter object with a quarter of 1 and a year of 2000 was returned.
- Severity: Low
- Priority: Medium
- Status: Passed
- Additional notes:

Case 4: testParseQuarter()

- Expected Parameters: Valid input string representing a quarter, e.g., "2001-Q4"
- Sending Parameters: input4 = "2001-Q4"
- Description: Testing the parsing of a valid input string into a Quarter object.
- Expected Result: The Quarter object with a quarter of 4 and a year of 2001.
- Actual Result: The Quarter object with a quarter of 4 and a year of 2001 was returned.
- Severity: Low
- Priority: Medium
- Status: Passed
- Additional notes:

Case 5: testParseQuarter()

- Expected Parameters: Valid input string representing a quarter, e.g., "Q3-1999"
- Sending Parameters: input5 = "Q3-1999"
- Description: Testing the parsing of a valid input string into a Quarter object.
- Expected Result: The Quarter object with a quarter of 3 and a year of 1999.
- Actual Result: The Quarter object with a quarter of 3 and a year of 1999 was returned.
- Severity: Low
- Priority: Medium
- Status: Passed
- Additional notes:

Overall Severity: Low

Overall Priority: Medium

Overall Status: Passed

Additional notes: All test cases passed, no bugs were found.