Bug reports

Quarter can go out of limits in `Quarter(int quarter, int year)` constructor

- **BR id**: #1
- **Tester**: Student
- **Date**: 2023-04-22
- **Description**: a new quarter is created even if the quarter provided in the constructor is < -1 or > 5
- **Method to be tested**: new Quarter(int quarter, int year)
- **Test cases**
    - `new Quarter(5, 2023)` (fails to throw `IllegalArgumentException`)
    - `new Quarter(-1, 2023)` (fails to throw `IllegalArgumentException`)

Quarter can go out of limits in `Quarter(int quarter, Year year)` constructor

- **BR id**: #2
- **Tester**: Student
- **Date**: 2023-04-22
- **Description**: same as #1
- **Method to be tested**: new Quarter(int quarter, int year)
- **Test cases**
    - `new Quarter(5, new Year(2023))` (fails to throw `IllegalArgumentException`)
    - `new Quarter(-1, new Year(2023))` (fails to throw `IllegalArgumentException`)

NullPointerException in `Quarter(int quarter, Year year)` constructor

- **BR id**: #3
- **Tester**: Student
- **Date**: 2023-04-22
- **Description**: providing `null` as the year object gives a `NullPointerException` that is raised from internal logic. Throwing `IllegalArgumentException` or handling null internally is required.
- **Method to be tested**: new Quarter(int quarter, Year year)
- **Test cases**
    - `new Quarter(3, null)` (fails with `NullPointerException`)

NullPointerException in `Quarter(Date date)` constructor

- **BR id**: #4
- **Tester**: Student
- **Date**: 2023-04-22

- **Description**: providing `null` as the date object gives a `NullPointerException` that is raised from internal logic. Throwing `IllegalArgumentException` or handling null internally is required.
- **Method to be tested**: `new Quarter(Date date)`
- **Test cases**
  - `new Quarter(null)` (fails with `NullPointerException`)

NullPointerException in `Quarter(Date date, TimeZone tz)` constructor

- **BR id**: #5
- **Tester**: Student
- **Date**: 2023-04-22
- **Description**: providing `null` as the tz object gives a `NullPointerException` that is raised from internal logic. Throwing `IllegalArgumentException` or handling null internally is required.
- **Method to be tested**: `new Quarter(Date date, TimeZone tz)`
- **Test cases**
  - `new Quarter(new Date(1672531200000l), null)` (fails with `NullPointerException`)

Quarter can go out of range in `parseQuarter(String quarter)`

- **BR id**: #6
- **Tester**: Student
- **Date**: 2023-04-22
- **Description**: providing a quarter that is < -1 or > 5 in the quarter string does not fail and a quarter is created. `IllegalArgumentException` should be thrown.
- **Method to be tested**: `Quarter.parseQuarter(String quarter)`
- **Test cases**
  - `Quarter.parseQuarter("2023 Q5")` (fails to throw `IllegalArgumentException`)
  - `Quarter.parseQuarter("2023-Q5")` (fails to throw `IllegalArgumentException`)
  - `Quarter.parseQuarter("2023/Q5")` (fails to throw `IllegalArgumentException`)
  - `Quarter.parseQuarter("Q5 2023")` (fails to throw `IllegalArgumentException`)
  - `Quarter.parseQuarter("Q5-2023")` (fails to throw `IllegalArgumentException`)
  - `Quarter.parseQuarter("Q5/2023")` (fails to throw `IllegalArgumentException`)

Comparing a Quarter to another class or to null is undocumented

- **BR id**: #7
- **Tester**: Student
- **Date**: 2023-04-23

- **Description**: a quarter turns out to be greater than null and greater than another object. This is undocumented.
- **Method to be tested**: `Quarter.compareTo(Object object)`
- **Test cases**
  - `(new Quarter()).compareTo(new Object()) > 0` (returns true. Can't determine whether this is a failure/success since it is undocumented)
  - `(new Quarter()).compareTo(null) > 0` (returns true. Can't determine whether this is a failure/success since it is undocumented)