Test.java:

```java
/**
 * date is a error exception class exrends Exception class
 *
 * @author Peng Gao (pgaooscar@gmail.com)
 * @version 0.1 November 11 2020
 */
class Error extends Exception {
  /**
   * java 11 defualt serial version ID
   */
  private static final long serialVersionUID = 1L;
  /**
   * code is error code
   */
  private int code;

  /**
   * constructor
   *
   * @param c is error code
   */
  public Error(int c) {
    code = c;
  }

  /**
   * get error code function
   *
   * @return error code
   */
  public int getCode() {
    return code;
  }
}

// To handle a date whose month is illegal
/**
 * ErrorMonth is a error exception class exrends Error class
 *
 * <pre>
 * ErrorMonth e1 = new ErrorMonth(1998, 4, 27);
 * </pre>
 */
```

```java
class ErrorMonth extends Error {
  /**
   * java 11 defualt serial version ID
   */
  private static final long serialVersionUID = 1L;
  /**
   * m is month
   */
  private int m;

  /**
   * constructor
   *
   * @param month is the month of date
   */
  public ErrorMonth(int month) {
    super(3);
    m = month;
  }

  // Call the getMonth(int) function defined in Question 1
  // to convert a numerical month into a string month
  /**
   * get month of date function
   *
   * @return month
   */
  public int getMonth() {
    return m;
  }

}

// To handle a date whose year is illegal
/**
 * ErrorYear is a error exception class exrends Error class
 *
 * <pre>
 * ErrorYear e1 = new ErrorYear(1998, 4, 27);
 * </pre>
 */
class ErrorYear extends Error {
  /**
   * java 11 defualt serial version ID
```

```java
   */
  private static final long serialVersionUID = 1L;
  /**
   * y is the year of date
   */
  private int y;

  /**
   * constructor
   *
   * @param year is the year of date
   */
  public ErrorYear(int year) {
    super(1);
    y = year;
  }

  /**
   * get year of date function
   *
   * @return y is the year of date
   */
  public int getYear() {
    return y;
  }
}

// Base class to handle illegal day
/**
 * ErrorDay is a error exception class exrends Error class
 *
 * <pre>
 * ErrorDay e1 = new ErrorDay(1998, 4, 27);
 * </pre>
 */
class ErrorDay extends Error {
  /**
   * java 11 defualt serial version ID
   */
  private static final long serialVersionUID = 1L;
  /**
   * m is the month of date
   */
  private int m;
```

```java
  /**
   * d is the day of date
   */
  private int d;

  /**
   * constructor
   *
   * @param month is the month of date
   * @param day   is the day of the date
   */
  public ErrorDay(int month, int day) {
    super(2);
    m = month;
    d = day;
  }

  // Call the global function getMonth(int) defined in Question 1 to re
turn
  // the string representing the month
  /**
   * get month of date function
   *
   * @return month
   */
  public int getMonth() {
    return m;
  }

  /**
   * get day of date function
   *
   * @return day
   */
  public int getDay() {
    return d;
  }
}

// To handle a date whose year is a leap year,
// month is Feburay, but day is not 29
/**
 * Error29Day is a error exception class exrends ErrorDay class
 *
```

```
 * <pre>
 * Error29Day e1 = new Error29Day(1996, 4, 19);
 * </pre>
 *
 * @author Weijian Xiong (im.xwjian@gmail.com)
 * @version 0.1.1 March 13 2020
 */
class Error29Day extends ErrorDay {
  /**
   * java 11 defualt serial version ID
   */
  private static final long serialVersionUID = 1L;
  /**
   * y is the year of date
   */
  private int y;

  /**
   * constructor
   *
   * @param month is the month of date
   * @param day   is the day of date
   * @param year  is the year of date
   */
  public Error29Day(int month, int day, int year) {

    super(month, day);
    y = year;
  }

  /**
   * get year of date function
   *
   * @return year of date
   */
  public int getYear() {
    return y;
  }
}

// To handle a date whose month is 1, or 3, or 7,
// or 8, or 10, or 12, but day is not 31
/**
 * Error31Day is a error exception class exrends ErrorDay class
```

```java
 *
 * <pre>
 * Error31Day e1 = new Error31Day(1998, 4, 27);
 * </pre>
 */
class Error31Day extends ErrorDay {
  /**
   * java 11 defualt serial version ID
   */
  private static final long serialVersionUID = 1L;
  /**
   * y is the year
   */
  private int y;

  /**
   * class constructor
   *
   * @param month is the month of date
   * @param day   is the day of date
   * @param year  is the year of date
   */
  public Error31Day(int month, int day, int year) {
    super(month, day);
    y = year;
  }

  /**
   * get year of date function
   *
   * @return y is the year of date
   */
  public int getYear() {
    return y;
  }
}

// To handle a date whose year is not a leap year,
// month is Feburay, but the day is greater than 28
/**
 * Error28Day is a error exception class exrends ErrorDay class
 *
 * <pre>
 * Error28Day e1 = new Error31Day(1998, 4, 27);
```

```
 * </pre>
 */
class Error28Day extends ErrorDay {
  /**
   * java 11 defualt serial version ID
   */
  private static final long serialVersionUID = 1L;
  /**
   * y is the year
   */
  private int y;

  /**
   * class constructor
   *
   * @param month is the month of date
   * @param day   is the day of date
   * @param year  is the year of date
   */
  public Error28Day(int month, int day, int year) {
    super(month, day);
    y = year;
  }

  /**
   * get year of date function
   *
   * @return y is the year of date
   */
  public int getYear() {
    return y;
  }
}

// To handle a date whose month is 4, or 6, or 9,
// or 11, but the day is greater than 30
/**
 * Error30Day is a error exception class exrends ErrorDay class
 *
 * <pre>
 * Error30Day e1 = new Error31Day(1998, 4, 27);
 * </pre>
 */
class Error30Day extends ErrorDay {
```

```java
  /**
   * serial
   */
  private static final long serialVersionUID = 1L;
  /**
   * y is the year
   */
  private int y;

  /**
   * constructor
   *
   * @param month is the month of date
   * @param day   is the day of date
   * @param year  is the year of date
   */
  public Error30Day(int month, int day, int year) {
    super(month, day);
    y = year;
  }

  /**
   * get year of date function
   *
   * @return y is the year of date
   */
  public int getYear() {
    return y;
  }
}

/**
 * Test is a class use to run date class
 *
 * <pre>
 * Date d1 = new Date(1998, 4, 27);
 * </pre>
 *
 * @author Peng Gao (pgaooscar@gmail.com)
 * @version 0.1 November 11 2020
 */

public class Test {
  /**
```

```
   * main class use to run the program
   *
   * @param args arguments
   */
  public static void main(String[] args) {
    // Error 21: Illegal day 34 for March
    System.out.println("--------------------------Test:: 1993, 3, 34--
------------------------");
    try {
      Date d1 = new Date(1993, 3, 34);
    } catch (Error29Day e3) {
      System.out.println("\nError " + e3.getCode() + ": 29th error " +
"\nIllegal:     Day: " + e3.getDay()
          + "  Month: " + e3.getMonth() + "  Year: " + e3.getYear());
    } catch (Error31Day e4) {
      System.out.println("\nError " + e4.getCode() + ": 31th error " +
"\nIllegal:     Day: " + e4.getDay() + " Month: "
          + e4.getMonth() + "  Year: " + e4.getYear());
    } catch (Error28Day e5) {
      System.out.println("\nError " + e5.getCode() + ": 28th error " +
"\nIllegal:     Day: " + e5.getDay() + " Month: "
          + e5.getMonth() + "  Year: " + e5.getYear());
    } catch (Error30Day e6) {
      System.out.println("\nError " + e6.getCode() + ": 30th error " +
"\nIllegal:     Day: " + e6.getDay() + " Month: "
          + e6.getMonth() + "  Year: " + e6.getYear());
    } catch (ErrorYear e) {
      System.out.println("\nError " + e.getCode() + ": year error" + "\
nIllegal:     Year: " + e.getYear());
    } catch (ErrorDay e1) {
      System.out.println("\nError " + e1.getCode() + ": day error" + "\
nIllegal:     Day: " + e1.getDay() + " Month: "
          + e1.getMonth());
    } catch (ErrorMonth e2) {
      System.out.println("\nError " + e2.getCode() + ": month error" +
"\nIllegal:     Month: " + e2.getMonth());
    }

    // Error 3: Unknown Month
    System.out.println("\n--------------------------Test:: 1993, -
1, 34--------------------------");
    try {
      Date d2 = new Date(1993, -1, 34);
    } catch (Error29Day e3) {
```

```
      System.out.println("\nError " + e3.getCode() + ": 29th error " +
"\nIllegal:     Day: " + e3.getDay()
          + "  Month: " + e3.getMonth() + "  Year: " + e3.getYear());
    } catch (Error31Day e4) {
      System.out.println("\nError " + e4.getCode() + ": 31th error " +
"\nIllegal:     Day: " + e4.getDay() + " Month: "
          + e4.getMonth() + "  Year: " + e4.getYear());
    } catch (Error28Day e5) {
      System.out.println("\nError " + e5.getCode() + ": 28th error " +
"\nIllegal:     Day: " + e5.getDay() + " Month: "
          + e5.getMonth() + "  Year: " + e5.getYear());
    } catch (Error30Day e6) {
      System.out.println("\nError " + e6.getCode() + ": 30th error " +
"\nIllegal:     Day: " + e6.getDay() + " Month: "
          + e6.getMonth() + "  Year: " + e6.getYear());
    } catch (ErrorYear e) {
      System.out.println("\nError " + e.getCode() + ": year error" + "\
nIllegal:     Year: " + e.getYear());
    } catch (ErrorDay e1) {
      System.out.println("\nError " + e1.getCode() + ": day error" + "\
nIllegal:     Day: " + e1.getDay() + " Month: "
          + e1.getMonth());
    } catch (ErrorMonth e2) {
      System.out.println("\nError " + e2.getCode() + ": month error" +
"\nIllegal:     Month: " + e2.getMonth());
    }

    // Error 22: Illegal day 29 for Feburary in 1993
    System.out.println("\n---------------------------
Test:: 1993, 2, 29---------------------------");
    try {
      Date d3 = new Date(1993, 2, 29);
    } catch (Error29Day e3) {
      System.out.println("\nError " + e3.getCode() + ": 29th error " +
"\nIllegal:     Day: " + e3.getDay()
          + "  Month: " + e3.getMonth() + "  Year: " + e3.getYear());
    } catch (Error31Day e4) {
      System.out.println("\nError " + e4.getCode() + ": 31th error " +
"\nIllegal:     Day: " + e4.getDay() + " Month: "
          + e4.getMonth() + "  Year: " + e4.getYear());
    } catch (Error28Day e5) {
      System.out.println("\nError " + e5.getCode() + ": 28th error " +
"\nIllegal:     Day: " + e5.getDay() + " Month: "
          + e5.getMonth() + "  Year: " + e5.getYear());
```

```
    } catch (Error30Day e6) {
      System.out.println("\nError " + e6.getCode() + ": 30th error " +
"\nIllegal:     Day: " + e6.getDay() + " Month: "
          + e6.getMonth() + "  Year: " + e6.getYear());
    } catch (ErrorYear e) {
      System.out.println("\nError " + e.getCode() + ": year error" + "\
nIllegal:     Year: " + e.getYear());
    } catch (ErrorDay e1) {
      System.out.println("\nError " + e1.getCode() + ": day error" + "\
nIllegal:     Day: " + e1.getDay() + " Month: "
          + e1.getMonth());
    } catch (ErrorMonth e2) {
      System.out.println("\nError " + e2.getCode() + ": month error" +
"\nIllegal:     Month: " + e2.getMonth());
    }

    // Error 3: Unknown Month
    System.out.println("\n---------------------------
Test:: 1993, 13, 25---------------------------");
    try {
      Date d4 = new Date(1993, 13, 25);
    } catch (Error29Day e3) {
      System.out.println("\nError " + e3.getCode() + ": 29th error " +
"\nIllegal:     Day: " + e3.getDay()
          + "  Month: " + e3.getMonth() + "  Year: " + e3.getYear());
    } catch (Error31Day e4) {
      System.out.println("\nError " + e4.getCode() + ": 31th error " +
"\nIllegal:     Day: " + e4.getDay() + " Month: "
          + e4.getMonth() + "  Year: " + e4.getYear());
    } catch (Error28Day e5) {
      System.out.println("\nError " + e5.getCode() + ": 28th error " +
"\nIllegal:     Day: " + e5.getDay() + " Month: "
          + e5.getMonth() + "  Year: " + e5.getYear());
    } catch (Error30Day e6) {
      System.out.println("\nError " + e6.getCode() + ": 30th error " +
"\nIllegal:     Day: " + e6.getDay() + " Month: "
          + e6.getMonth() + "  Year: " + e6.getYear());
    } catch (ErrorYear e) {
      System.out.println("\nError " + e.getCode() + ": year error" + "\
nIllegal:     Year: " + e.getYear());
    } catch (ErrorDay e1) {
      System.out.println("\nError " + e1.getCode() + ": day error" + "\
nIllegal:     Day: " + e1.getDay() + " Month: "
          + e1.getMonth());
```

```
    } catch (ErrorMonth e2) {
      System.out.println("\nError " + e2.getCode() + ": month error" +
"\nIllegal:     Month: " + e2.getMonth());
    }

    // Error 21: Illegal day 32 for December
    System.out.println("\n---------------------------
Test:: 1988, 12, 32---------------------------");
    try {
      Date d5 = new Date(1988, 12, 32);
    } catch (Error29Day e3) {
      System.out.println("\nError " + e3.getCode() + ": 29th error " +
"\nIllegal:     Day: " + e3.getDay()
          + "  Month: " + e3.getMonth() + "  Year: " + e3.getYear());
    } catch (Error31Day e4) {
      System.out.println("\nError " + e4.getCode() + ": 31th error " +
"\nIllegal:     Day: " + e4.getDay() + " Month: "
          + e4.getMonth() + "  Year: " + e4.getYear());
    } catch (Error28Day e5) {
      System.out.println("\nError " + e5.getCode() + ": 28th error " +
"\nIllegal:     Day: " + e5.getDay() + " Month: "
          + e5.getMonth() + "  Year: " + e5.getYear());
    } catch (Error30Day e6) {
      System.out.println("\nError " + e6.getCode() + ": 30th error " +
"\nIllegal:     Day: " + e6.getDay() + " Month: "
          + e6.getMonth() + "  Year: " + e6.getYear());
    } catch (ErrorYear e) {
      System.out.println("\nError " + e.getCode() + ": year error" + "\
nIllegal:     Year: " + e.getYear());
    } catch (ErrorDay e1) {
      System.out.println("\nError " + e1.getCode() + ": day error" + "\
nIllegal:     Day: " + e1.getDay() + " Month: "
          + e1.getMonth());
    } catch (ErrorMonth e2) {
      System.out.println("\nError " + e2.getCode() + ": month error" +
"\nIllegal:     Month: " + e2.getMonth());
    }

    // Error 24: Illegal day 31 for April
    System.out.println("\n---------------------------
Test:: 1993, 4, 31---------------------------");
    try {
      Date d6 = new Date(1993, 4, 31);
    } catch (Error29Day e3) {
```

```
      System.out.println("\nError " + e3.getCode() + ": 29th error " +
"\nIllegal:     Day: " + e3.getDay()
          + "  Month: " + e3.getMonth() + "  Year: " + e3.getYear());
    } catch (Error31Day e4) {
      System.out.println("\nError " + e4.getCode() + ": 31th error " +
"\nIllegal:     Day: " + e4.getDay() + " Month: "
          + e4.getMonth() + "  Year: " + e4.getYear());
    } catch (Error28Day e5) {
      System.out.println("\nError " + e5.getCode() + ": 28th error " +
"\nIllegal:     Day: " + e5.getDay() + " Month: "
          + e5.getMonth() + "  Year: " + e5.getYear());
    } catch (Error30Day e6) {
      System.out.println("\nError " + e6.getCode() + ": 30th error " +
"\nIllegal:     Day: " + e6.getDay() + " Month: "
          + e6.getMonth() + "  Year: " + e6.getYear());
    } catch (ErrorYear e) {
      System.out.println("\nError " + e.getCode() + ": year error" + "\
nIllegal:     Year: " + e.getYear());
    } catch (ErrorDay e1) {
      System.out.println("\nError " + e1.getCode() + ": day error" + "\
nIllegal:     Day: " + e1.getDay() + " Month: "
          + e1.getMonth());
    } catch (ErrorMonth e2) {
      System.out.println("\nError " + e2.getCode() + ": month error" +
"\nIllegal:     Month: " + e2.getMonth());
    }

    // Error 3 Unknown Month
    System.out.println("\n---------------------------
Test:: 1993, 13, 25---------------------------");
    try {
      Date d7 = new Date(1993, 13, 25);
    } catch (Error29Day e3) {
      System.out.println("\nError " + e3.getCode() + ": 29th error " +
"\nIllegal:     Day: " + e3.getDay()
          + "  Month: " + e3.getMonth() + "  Year: " + e3.getYear());
    } catch (Error31Day e4) {
      System.out.println("\nError " + e4.getCode() + ": 31th error " +
"\nIllegal:     Day: " + e4.getDay() + " Month: "
          + e4.getMonth() + "  Year: " + e4.getYear());
    } catch (Error28Day e5) {
      System.out.println("\nError " + e5.getCode() + ": 28th error " +
"\nIllegal:     Day: " + e5.getDay() + " Month: "
          + e5.getMonth() + "  Year: " + e5.getYear());
```

```
    } catch (Error30Day e6) {
      System.out.println("\nError " + e6.getCode() + ": 30th error " +
"\nIllegal:     Day: " + e6.getDay() + " Month: "
          + e6.getMonth() + "  Year: " + e6.getYear());
    } catch (ErrorYear e) {
      System.out.println("\nError " + e.getCode() + ": year error" + "\
nIllegal:     Year: " + e.getYear());
    } catch (ErrorDay e1) {
      System.out.println("\nError " + e1.getCode() + ": day error" + "\
nIllegal:     Day: " + e1.getDay() + " Month: "
          + e1.getMonth());
    } catch (ErrorMonth e2) {
      System.out.println("\nError " + e2.getCode() + ": month error" +
"\nIllegal:     Month: " + e2.getMonth());
    }

    // Error 24: Illegal day 31 for November
    System.out.println("\n---------------------------
Test:: 1993, 11, 31---------------------------");
    try {
      Date d8 = new Date(1993, 11, 31);
    } catch (Error29Day e3) {
      System.out.println("\nError " + e3.getCode() + ": 29th error " +
"\nIllegal:     Day: " + e3.getDay()
          + "  Month: " + e3.getMonth() + "  Year: " + e3.getYear());
    } catch (Error31Day e4) {
      System.out.println("\nError " + e4.getCode() + ": 31th error " +
"\nIllegal:     Day: " + e4.getDay() + " Month: "
          + e4.getMonth() + "  Year: " + e4.getYear());
    } catch (Error28Day e5) {
      System.out.println("\nError " + e5.getCode() + ": 28th error " +
"\nIllegal:     Day: " + e5.getDay() + " Month: "
          + e5.getMonth() + "  Year: " + e5.getYear());
    } catch (Error30Day e6) {
      System.out.println("\nError " + e6.getCode() + ": 30th error " +
"\nIllegal:     Day: " + e6.getDay() + " Month: "
          + e6.getMonth() + "  Year: " + e6.getYear());
    } catch (ErrorYear e) {
      System.out.println("\nError " + e.getCode() + ": year error" + "\
nIllegal:     Year: " + e.getYear());
    } catch (ErrorDay e1) {
      System.out.println("\nError " + e1.getCode() + ": day error" + "\
nIllegal:     Day: " + e1.getDay() + " Month: "
          + e1.getMonth());
```

```
    } catch (ErrorMonth e2) {
      System.out.println("\nError " + e2.getCode() + ": month error" +
"\nIllegal:     Month: " + e2.getMonth());
    }

  }
}
```

Date.java:

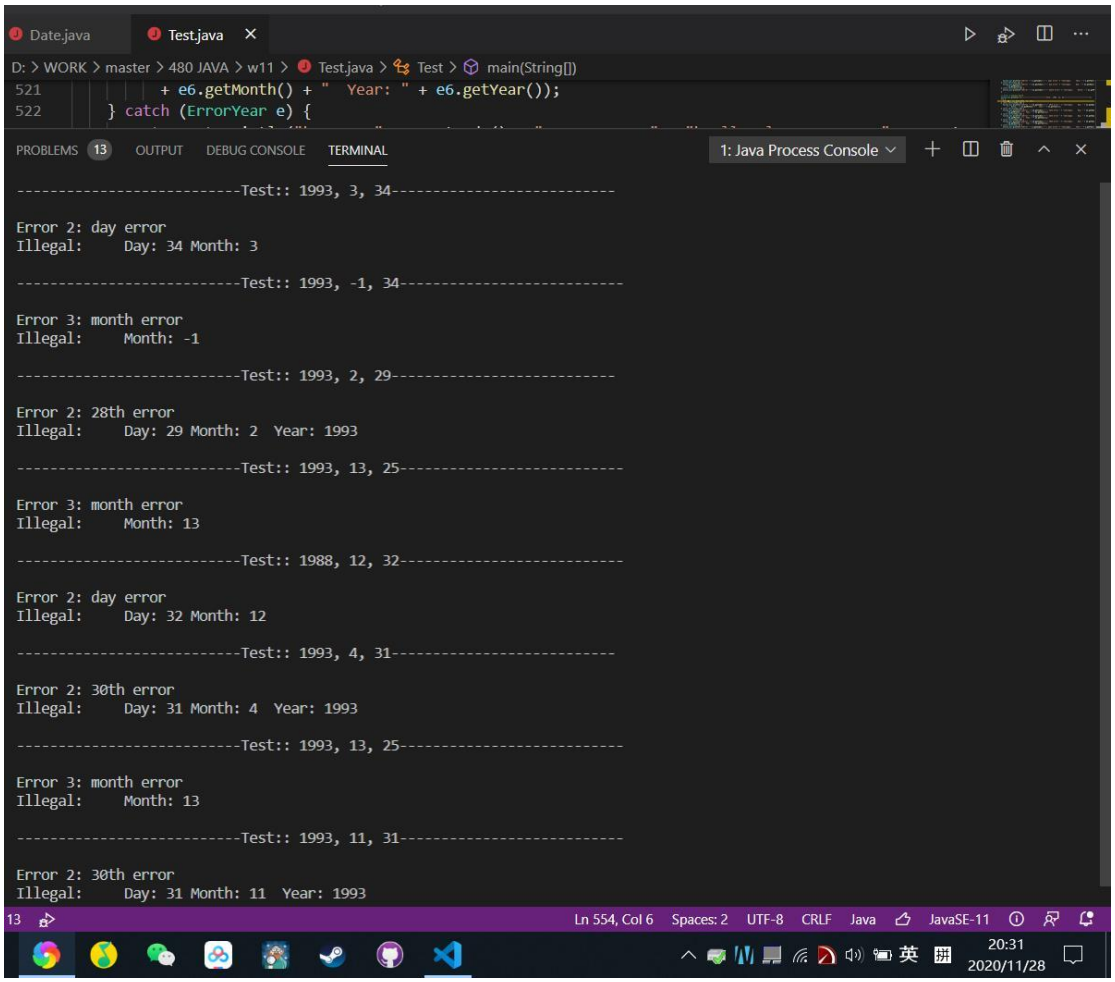
```
//                        --------------+
//                        | Error       |
//                        |   int code  |
//                        --------------+
//                              |
//                              | public
//                              |
//     +-----------------------+------------------+
//     |                       |                  |
//     | public                | public           | public
//     |                       |                  |
// +--------------+        +------------+     +------------+
// | ErroYear     |        | ErrorDay   |     | ErrorMonth |
// |   int y      |        |  int m     |     |   int m    |
// |              |        |  int d     |     |            |
// +--------------+        +----+-------+     +------------+
//     code:1                    | code: 2         code: 3
//                               |
//                +-------------+
//                |public       | public
//                |             |
//         +------+------+ +----+-------+
//         | Error31Days | | Error29Day |
//         |             | |   int y    |
//         +-------------+ +------------+
//            code: 21         code: 22

/**
 * Date is a date class
 *
```

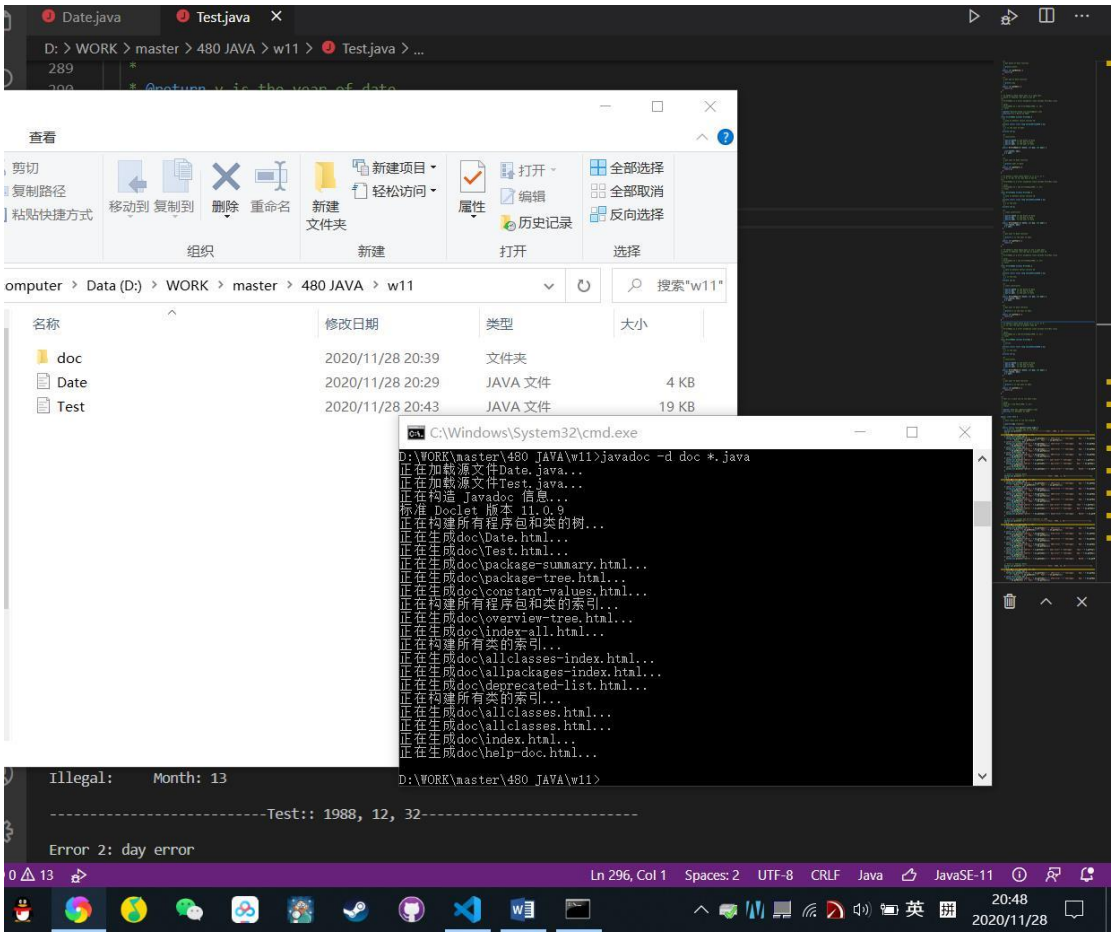
```java
 * <pre>
 * Date d1 = new Date(1998, 4, 27);
 * </pre>
 *
 * @author Peng Gao (pgaooscar@gmail.com)
 * @version 0.1 November 11 2020
 */
public class Date {
    /**
     * y is year of date
     */
    private int y;
    /**
     * m is month of date
     */
    private int m;
    /**
     * d is day of date
     */
    private int d;
    /**
     * leap_yeay_flag is using to check does it is a leap year
     */
    private int leap_year_flag;

    /**
     * date class constructor
     *
     * @param y1 is the year of new date
     * @param m1 is the month of new date
     * @param d1 is the day of new date
     * @throws Error29Day error exception
     * @throws Error31Day error exception
     * @throws Error28Day error exception
     * @throws Error30Day error exception
     * @throws ErrorDay   error exception
     * @throws ErrorMonth error exception
     * @throws ErrorYear  error exception
     *
     */
    public Date(int y1, int m1, int d1)
            throws Error29Day, Error31Day, Error28Day, Error30Day, Erro
rDay, ErrorMonth, ErrorYear {
```

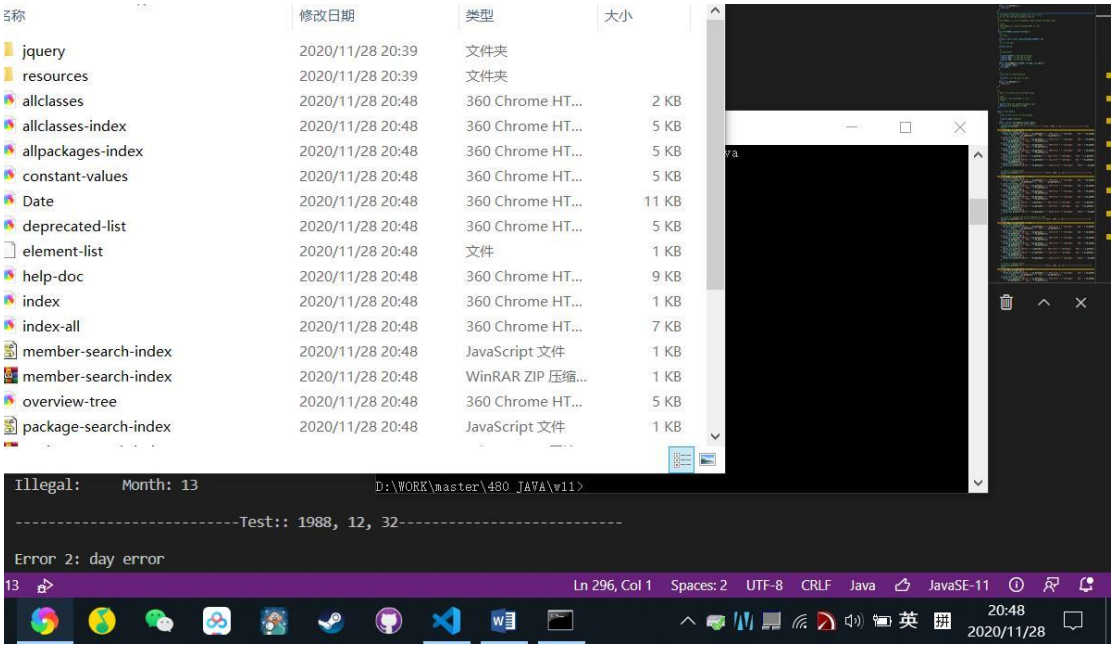
```
        if (y1 > 9999 || y1 < 0) {
            throw new ErrorYear(y1);
        } else if (m1 < 0 || m1 > 12) {
            throw new ErrorMonth(m1);
        } else if (d1 < 0 || d1 > 31) {
            throw new ErrorDay(m1, d1);
        } else if (y1 % 4 != 0 && m1 == 2 && d1 > 28) {
            throw new Error28Day(m1, d1, y1);
        } else if (y1 % 4 == 0 && m1 == 2 && d1 > 29) {
            throw new Error29Day(m1, d1, y1);
        } else if (m1 == 4 || m1 == 6 || m1 == 9 || m1 == 11) {
            if (d1 > 30) {
                throw new Error30Day(m1, d1, y1);
            }
        } else if (m1 == 1 || m1 == 3 || m1 == 7 || m1 == 8 || m1 == 10
|| m1 == 12) {
            if (d1 > 31) {
                throw new Error31Day(m1, d1, y1);
            }
        } else {
            y = y1;
            m = m1;
            d = d1;
            leap_year_flag = 1;
        }
    }

    // Call global function "isLeapYear(int)"
    // to check if it is a leap year
    /**
     * check if is leap year or not function
     *
     * @param year is the year you want to check
     * @return 1 or 0
     */
    public int isLeapYear(int year) {
        if (year % 4 == 0) {
            leap_year_flag = 1;
            return leap_year_flag;
        } else {
            leap_year_flag = 0;
            return leap_year_flag;
        }
    }
```

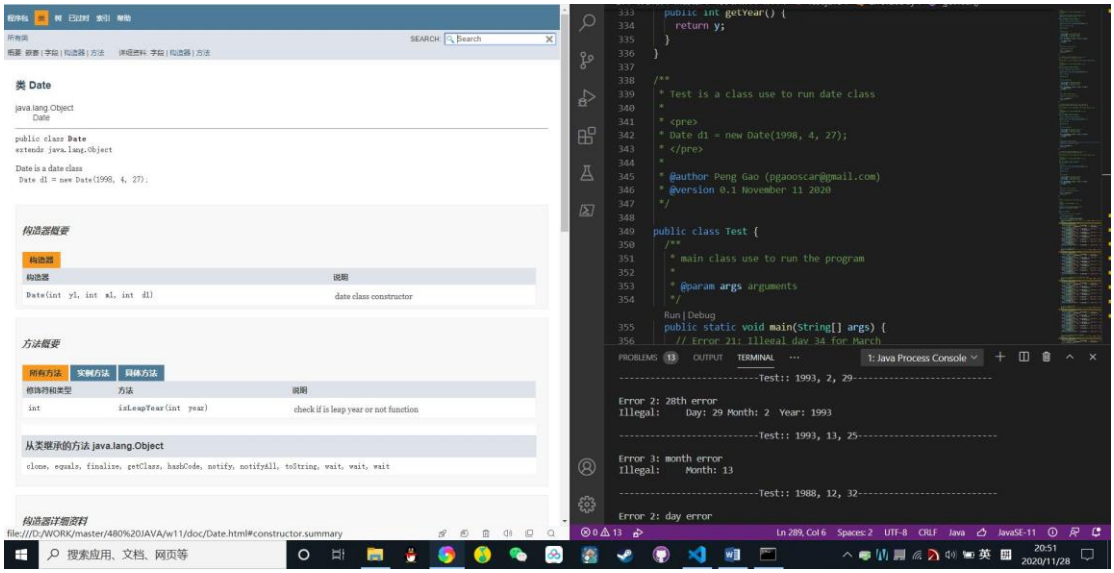
```
}
```

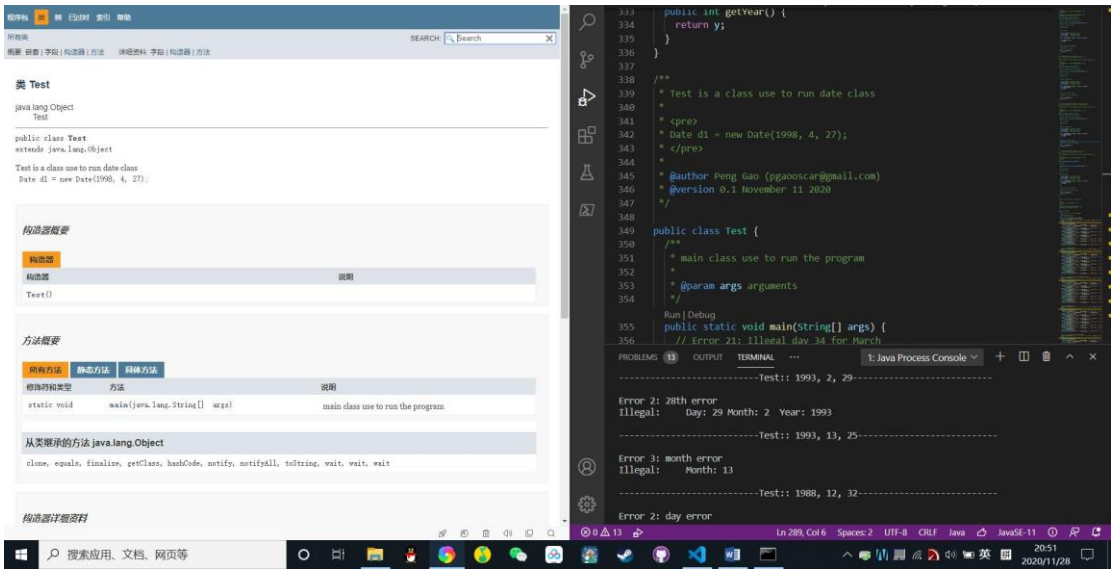
Output & Javadoc:




Date.java
Test.java
D: > WORK > master > 480 JAVA > w11 > Test.java > ...
查看
剪切
复制路径
粘贴快捷方式
移动到 复制到 删除 重命名
新建文件夹
新建项目
轻松访问
属性
打开
编辑
历史记录
全部选择
全部取消
反向选择
组织
新建
打开
选择
omputer > Data (D:) > WORK > master > 480 JAVA > w11
搜索"w11"
名称 修改日期 类型 大小
doc 2020/11/28 20:39 文件夹
Date 2020/11/28 20:29 JAVA 文件 4 KB
Test 2020/11/28 20:43 JAVA 文件 19 KB
C:\Windows\System32\cmd.exe
D:\WORK\master\480 JAVA\w11>javadoc -d doc *.java
正在加载源文件Date.java...
正在加载源文件Test.java...
正在构造 Javadoc 信息...
标准 Doclet 版本 11.0.9
正在构建所有程序包和类的树...
正在生成doc\Date.html...
正在生成doc\Test.html...
正在生成doc\package-summary.html...
正在生成doc\package-tree.html...
正在生成doc\constant-values.html...
正在构建所有程序包和类的索引...
正在生成doc\overview-tree.html...
正在生成doc\index-all.html...
正在构建所有类的索引...
正在生成doc\allclasses-index.html...
正在生成doc\allpackages-index.html...
正在生成doc\deprecated-list.html...
正在构建所有类的索引...
正在生成doc\allclasses.html...
正在生成doc\allclasses.html...
正在生成doc\index.html...
正在生成doc\help-doc.html...
D:\WORK\master\480 JAVA\w11>
Illegal: Month: 13
----------Test:: 1988, 12, 32----------
Error 2: day error
Ln 296, Col 1 Spaces: 2 UTF-8 CRLF Java JavaSE-11
20:48
2020/11/28


名称 修改日期 类型 大小
jquery 2020/11/28 20:39 文件夹
resources 2020/11/28 20:39 文件夹
allclasses 2020/11/28 20:48 360 Chrome HT... 2 KB
allclasses-index 2020/11/28 20:48 360 Chrome HT... 5 KB
allpackages-index 2020/11/28 20:48 360 Chrome HT... 5 KB
constant-values 2020/11/28 20:48 360 Chrome HT... 5 KB
Date 2020/11/28 20:48 360 Chrome HT... 11 KB
deprecated-list 2020/11/28 20:48 360 Chrome HT... 5 KB
element-list 2020/11/28 20:48 文件 1 KB
help-doc 2020/11/28 20:48 360 Chrome HT... 9 KB
index 2020/11/28 20:48 360 Chrome HT... 1 KB
index-all 2020/11/28 20:48 360 Chrome HT... 7 KB
member-search-index 2020/11/28 20:48 JavaScript 文件 1 KB
member-search-index 2020/11/28 20:48 WinRAR ZIP 压缩... 1 KB
overview-tree 2020/11/28 20:48 360 Chrome HT... 5 KB
package-search-index 2020/11/28 20:48 JavaScript 文件 1 KB
D:\WORK\master\480 JAVA\w11>
Illegal: Month: 13
----------Test:: 1988, 12, 32----------
Error 2: day error
Ln 296, Col 1 Spaces: 2 UTF-8 CRLF Java JavaSE-11
20:48
2020/11/28


类 Date
java.lang.Object
Date
public class Date
extends java.lang.Object
Date is a date class
Date d1 = new Date(1998, 4, 27);
构造器概要
构造器
构造器
说明
Date(int y1, int m1, int d1)
date class constructor
方法概要
所有方法
实例方法
具体方法
修饰符和类型
方法
说明
int
isLeapYear(int year)
check if is leap year or not function
从类继承的方法 java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
构造器详细资料
file:///D:/WORK/master/480%20JAVA/w11/doc/Date.html#constructor.summary
public int getYear() {
return y;
}
/**
* Test is a class use to run date class
*
* <pre>
* Date d1 = new Date(1998, 4, 27);
* </pre>
*
* @author Peng Gao (pgaooscar@gmail.com)
* @version 0.1 November 11 2020
*/
public class Test {
/**
* main class use to run the program
*
* @param args arguments
*/
Run | Debug
public static void main(String[] args) {
// Error 21: Illegal day 34 for March
PROBLEMS 13
OUTPUT
TERMINAL
1: Java Process Console
----------Test:: 1993, 2, 29----------
Error 2: 28th error
Illegal: Day: 29 Month: 2 Year: 1993
----------Test:: 1993, 13, 25----------
Error 3: month error
Illegal: Month: 13
----------Test:: 1988, 12, 32----------
Error 2: day error
Ln 289, Col 6
Spaces: 2
UTF-8
CRLF
Java
JavaSE-11
搜索应用、文档、网页等
20:51
2020/11/28


类 Test
java.lang.Object
Test
public class Test
extends java.lang.Object
Test is a class use to run date class
Date d1 = new Date(1998, 4, 27);
构造器概要
构造器
构造器
说明
Test()
方法概要
所有方法
静态方法
具体方法
修饰符和类型
方法
说明
static void
main(java.lang.String[] args)
main class use to run the program
从类继承的方法 java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
构造器详细资料
public int getYear() {
return y;
}
/**
* Test is a class use to run date class
*
* <pre>
* Date d1 = new Date(1998, 4, 27);
* </pre>
*
* @author Peng Gao (pgaooscar@gmail.com)
* @version 0.1 November 11 2020
*/
public class Test {
/**
* main class use to run the program
*
* @param args arguments
*/
Run | Debug
public static void main(String[] args) {
// Error 21: Illegal day 34 for March
PROBLEMS 13
OUTPUT
TERMINAL
1: Java Process Console
----------Test:: 1993, 2, 29----------
Error 2: 28th error
Illegal: Day: 29 Month: 2 Year: 1993
----------Test:: 1993, 13, 25----------
Error 3: month error
Illegal: Month: 13
----------Test:: 1988, 12, 32----------
Error 2: day error
Ln 289, Col 6
Spaces: 2
UTF-8
CRLF
Java
JavaSE-11
搜索应用、文档、网页等
20:51
2020/11/28