

Assignment 4

Problem Description:

Define a class named Time for encapsulating a time. The class contains the following:

1. A data field of the long time that stores the elapsed time since midnight, Jan 1, 1970.
2. A no-arg constructor that constructs a Time for the current time.
3. A constructor with the specified hour, minute, and second to create a Time.
4. A constructor with the specified elapsed time since midnight, Jan 1, 1970.
5. The getHour() method that returns the current hour in the range 0-23.
6. The getMinute() method that returns the current minute in the range 0-59.
7. The getSecond() method that returns the current second in the range 0-59.
8. The getSeconds() method that returns the elapsed total seconds.
9. The toString() method that returns a string such as "1 hour 2 minutes 1 second" and "14 hours 21 minutes 1 second".
10. Implement the Comparable<Time> interface to compare this Time with another one based on their elapse seconds. The compareTo method returns the difference between this object's elapse seconds and the another's.
11. Implement the Cloneable interface to clone a Time object.

Write a test program that produces the following sample run:

<Sample Run>

Enter time1 (hour minute second): 331 34 674 <Enter>

19 hours 45 minutes 14 seconds

Elapsed seconds in time1: 1194314

Enter time2 (elapsed time): 93889345 <Enter>

16 hours 22 minutes 25 seconds

Elapsed seconds in time2: 93889345

time1.compareTo(time2)? -92695031

time3 is created as a clone of time1

time1.compareTo(time3)? 0

<End Sample Run>

<Sample Run>

Enter time1 (hour minute second): 1 2 3 <Enter>

1 hour 2 minutes 3 seconds

Elapsed seconds in time1: 3723

Enter time2 (elapsed time): 193032 <Enter>

5 hours 37 minutes 12 seconds

Elapsed seconds in time2: 193032

time1.compareTo(time2)? -189309

time3 is created as a clone of time1

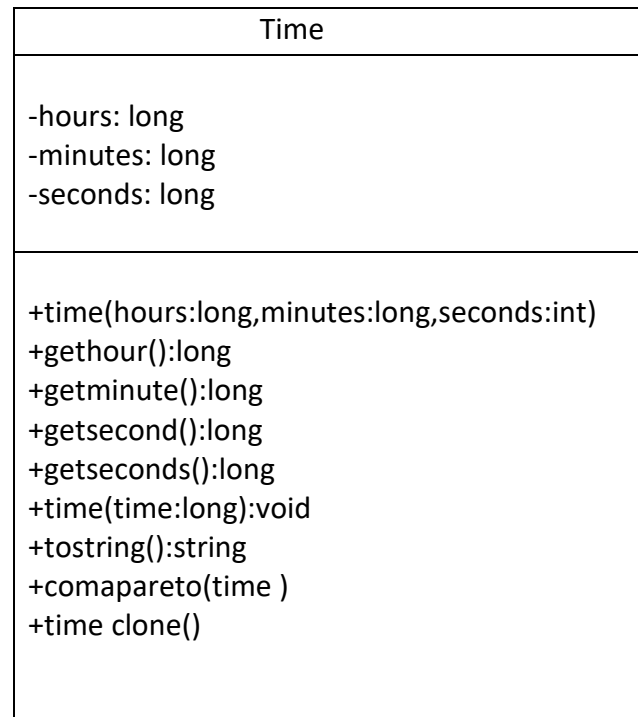
```
time1.compareTo(time3)? 0  
<End Sample Run>
```

Analysis:

- In the given problem we have to create class named time for encapsulating time.
This class contains
 1. A data field of long time that stores the elapsed time since midnight, Jan 1, 1970.
 2. No argument constructor that constructs time for current time.
 3. Constructor with specified hour, minute and seconds to create a time.
 4. Constructor with specified elapsed time since midnight jan 1,1970.
 5. getHour() method that returns the current hour in the range 0-23
 6. getMinute() method that returns the current minute in range 0-59
 7. getSecond () method that returns the current second in range 0-59
 8. getSeconds () returns elapsed seconds.
 9. The toString method that return a string
 10. Implement the comparable<time> interface to compare this tme with another one based on their elapse seconds.
 11. The compare to method must return the difference between this objects elapsed seconds and the another's
 12. Also implement the cloneable interface to clone the time object.
- We have to put the sample inputs from the user and calculate the elapsed time and display results.

Design:

UML Diagram for the problem



Coding:

```
/* Basavraj Jaliminche
 * id- 8800149
 * Assignment 4
 * PROG8580-Computer Programming
 */

package Assignment4;

//here time class implements comparable and cloneable
public class time implements Comparable<time>, Cloneable {
    private long t;

    public time() {
        t = System.currentTimeMillis();
    }

    // constructor spacificied with hour,minute,seconds
    public time(long hr, long min, long sec) {
        t = (((hr * 60) + min) * 60) + sec) * 1000;
    }

    public int getHour() {
        // hours in the between 0 to 23.
        return (int) (t / (1000 * 60 * 60)) % 24;
    }

    public int getMinute() {
        // minutes between 0 to 59
        return (int) (t / (1000 * 60)) % 60;
    }

    public int getSecond() {
        // seconds between 0 to 59.
        return (int) (t / 1000) % 60;
    }

    public int getSeconds() {
        // returns the elapsed time seconds.
        return (int) (t / 1000);
    }

    public time(long time) {
        t = time * 1000;
    }
}
```

```

@Override
public String toString() {
    return String.format("%d hours %d minutes %d seconds", getHour(), getMinute(), getSecond());
}

@Override
public int compareTo(time timeCompare) {
    return (int) (this.getSeconds() - timeCompare.getSeconds());
}

// time clone throws exception
public time Clone() throws CloneNotSupportedException {
    return (time) super.clone();
}
}

/* Basavraj Jaliminche
 * id- 8800149
 * Assignment 4
 * PROG8580-Computer Programming
 */

package Assignment4;

import java.util.Scanner;

public class encasptime {

    public static void main(String[] args) throws CloneNotSupportedException {

        // scanner class
        Scanner sc = new Scanner(System.in);

        // defining variables
        int hrs, mins, secs;

        // Taking input from the user

        System.out.print("<sample run>\n");
        System.out.print("Enter time1 (hour minute second): ");

        hrs = sc.nextInt();
        mins = sc.nextInt();
        secs = sc.nextInt();

        time Time1 = new time(hrs, mins, secs);

        // for taking output used toString method

```

```
System.out.println(Time1.toString());

System.out.println("Elapsed seconds in Time 1: " + Time1.getSeconds());

System.out.println("Enter Time 2 (Elapsed Time): ");

long mills = sc.nextInt();

time Time2 = new time(mills);

System.out.println(Time2);

System.out.println("Elapsed seconds in Time 2: " + Time2.getSeconds());

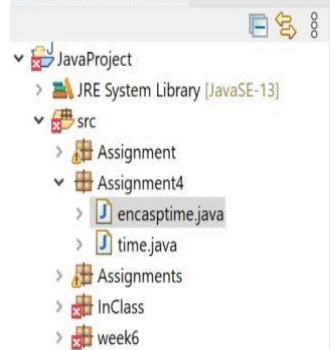
System.out.println("Time 1.compareTo(Time 2)? " + Time1.compareTo(Time2));

time T3 = Time1.Clone();

System.out.println("Time 3 is created as a clone of Time 1 Time1.compareTo(Time 3)? " + Time1.compareTo(T3));
System.out.print("<End Saple Run>");
sc.close();
}
}
```



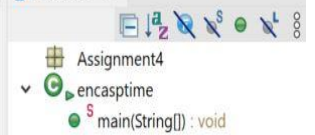
Package Explorer



time.java encasptime.java

```
1 /* Basavraj Jaliminche
2  * id- 8800149
3  * Assignment 4
4  * PROG8580-Computer Programming
5  */
6
7 package Assignment4;
8
9 import java.util.Scanner;
10
11 public class encasptime {
12
13     public static void main(String[] args) throws CloneNotSupportedException {
14
15         // scanner class
16         Scanner sc = new Scanner(System.in);
17
18         // defining variables
19         int hrs, mins, secs;
20
21         // Taking input from the user
22
23         System.out.print("<sample run>\n");
24         System.out.print("Enter time1 (hour minute second): ");
25
26         hrs = sc.nextInt();
27         mins = sc.nextInt();
28         secs = sc.nextInt();
29
30         time Time1 = new time(hrs, mins, secs);
31
32         // for taking output used toString method
33         System.out.println(Time1.toString());
34
35         System.out.println("Elapsed seconds in Time 1: " + Time1.getSeconds());
36
37         System.out.println("Enter Time 2 (Elapsed Time): ");
38
39         long mills = sc.nextInt();
40
```

Outline



Problems @ Javadoc Declaration Console

<terminated> encasptime [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Apr 8, 2022, 6:56:34 AM – 6:56:55 AM)

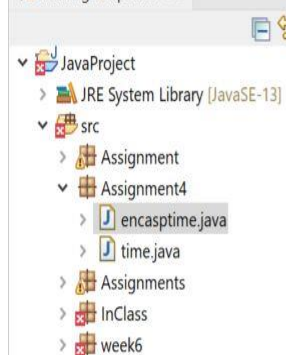
Writable

Smart Insert

9:26:147



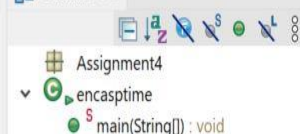
Package Explorer



time.java encasptime.java

```
18 // defining variables
19 int hrs, mins, secs;
20
21 // Taking input from the user
22
23 System.out.print("<sample run>\n");
24 System.out.print("Enter time1 (hour minute second): ");
25
26 hrs = sc.nextInt();
27 mins = sc.nextInt();
28 secs = sc.nextInt();
29
30 time Time1 = new time(hrs, mins, secs);
31
32 // for taking output used toString method
33 System.out.println(Time1.toString());
34
35 System.out.println("Elapsed seonds in Time 1: " + Time1.getSeconds());
36
37 System.out.println("Enter Time 2 (Elapsed Time): ");
38
39 long mills = sc.nextInt();
40
41 time Time2 = new time(mills);
42
43 System.out.println(Time2);
44
45 System.out.println("Elapsed seconds in Time 2: " + Time2.getSeconds());
46
47 System.out.println("Time 1.compareTo(Time 2)? " + Time1.compareTo(Time2));
48
49 time T3 = Time1.Clone();
50
51 System.out.println("Time 3 is created as a clone of Time 1 Time1.compareTo(Time 3)? " + Time1.compareTo(T3));
52 System.out.print("<End Saple Run>");
53 sc.close();
54 }
55 }
56 }
57
```

Outline



Problems @ Javadoc Declaration Console

<terminated> encasptime [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Apr 8, 2022, 6:56:34 AM – 6:56:55 AM)

Writable

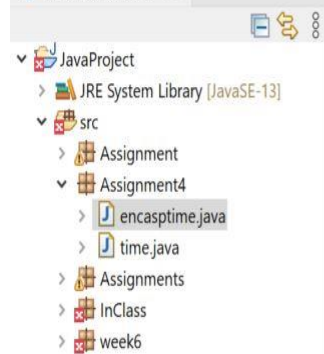
Smart Insert

9 : 26 : 147

7:10 AM



Package Explorer



time.java encasptime.java

```
1 /* Basavraj Jaliminche
2  * id- 8800149
3  * Assignment 4
4  * PROG8580-Computer Programming
5  */
6
7 package Assignment4;
8
9 //here time class implements comparable and clonable
10 public class time implements Comparable<time>, Cloneable {
11     private long t;
12
13     public time() {
14         t = System.currentTimeMillis();
15     }
16
17     // constructor spacificied with hour,minute,seconds
18     public time(long hr, long min, long sec) {
19         t = (((hr * 60) + min) * 60) + sec * 1000;
20     }
21
22
23     public int getHour() {
24         // hours in the between 0 to 23.
25         return (int) (t / (1000 * 60 * 60)) % 24;
26     }
27
28     public int getMinute() {
29         // minutes between 0 to 59
30         return (int) (t / (1000 * 60)) % 60;
31     }
32
33     public int getSecond() {
34         // seconds between 0 to 59.
35         return (int) (t / 1000) % 60;
36     }
37
38     public int getSeconds() {
39         // returns the elapsed time seconds.
40         return (int) (t / 1000);
41     }
42 }
```

Outline



Problems @ Javadoc Declaration Console

<terminated> encasptime [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Apr 8, 2022, 6:56:34 AM – 6:56:55 AM)

Writable

Smart Insert

53 : 45 : 1131



Package Explorer

JavaProject

- JRE System Library [JavaSE-13]
- src
 - Assignment
 - Assignment4
 - encasptime.java
 - time.java
 - Assignments
 - InClass
 - week6

time.java encasptime.java

```
24 // hours in the between 0 to 23.
25 return (int) (t / (1000 * 60 * 60)) % 24;
26 }
27
28 public int getMinute() {
29 // minutes between 0 to 59
30 return (int) (t / (1000 * 60)) % 60;
31 }
32
33 public int getSecond() {
34 // seconds between 0 to 59.
35 return (int) (t / 1000) % 60;
36 }
37
38 public int getSeconds() {
39 // returns the elapsed time seconds.
40 return (int) (t / 1000);
41 }
42
43 public time(long time) {
44 t = time * 1000;
45 }
46
47 @Override
48 public String toString() {
49 return String.format("%d hours %d minutes %d seconds", getHour(), getMinute(), getSecond());
50 }
51
52 @Override
53 public int compareTo(time timeCompare) {
54 return (int) (this.getSeconds() - timeCompare.getSeconds());
55 }
56
57 // time clone throws exception
58 public time Clone() throws CloneNotSupportedException {
59 return (time) super.clone();
60 }
61
62 }
63
```

Outline

Assignment4

- time
 - t : long
 - time()
 - time(long, long, long)
 - getHour() : int
 - getMinute() : int
 - getSecond() : int
 - getSeconds() : int
 - time(long)
 - toString() : String
 - compareTo(time) : int
 - Clone() : time

Problems @ Javadoc Declaration Console

<terminated> encasptime [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Apr 8, 2022, 6:56:34 AM – 6:56:55 AM)

Writable

Smart Insert

53 : 45 : 1131

Testing:

Test 1:

Java - JavaProject/src/Assignment4/encasptime.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

The screenshot displays the Eclipse IDE interface. The Package Explorer on the left shows the project structure: JavaProject > JRE System Library [JavaSE-13] > src > Assignment4 > encasptime.java. The main editor shows the code for `encasptime.java`, which includes a `main` method that takes user input for hours, minutes, and seconds, calculates elapsed time, and prints the results. The console at the bottom shows the output of the program, including sample input and output for two different time intervals.

```
12 public static void main(String[] args) throws CloneNotSupportedException {
13
14     // scanner class
15     Scanner sc = new Scanner(System.in);
16
17     // defining variables
18     int hrs, mins, secs;
19
20     // Taking input from the user
21
22     System.out.print("<sample run>\n");
23     System.out.print("Enter time1 (hour minute second): ");
24
25
26     hrs = sc.nextInt();
27     mins = sc.nextInt();
28     secs = sc.nextInt();
29
30     time Time1 = new time(hrs, mins, secs);
31
32     // for taking output used toString method
33     System.out.println(Time1.toString());
34
35     System.out.println("Elapsed seonds in Time 1: " + Time1.getSeconds());
36 }
```

Problems @ Javadoc Declaration Console

<terminated> encasptime [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Apr 8, 2022, 6:55:17 AM – 6:55:50 AM)

<sample run>
Enter time1 (hour minute second): 331 34 674
19 hours 45 minutes 14 seconds
Elapsed seonds in Time 1: 1194314
Enter Time 2 (Elapsed Time):
93889345
16 hours 22 minutes 25 seconds
Elapsed seconds in Time 2: 93889345
Time 1.compareTo(Time 2)? -92695031
Time 3 is created as a clone of Time 1 Time1.compareTo(Time 3)? 0
<End Saple Run>

Test2:

Java - JavaProject/src/Assignment4/encasptime.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

The screenshot displays the Eclipse IDE interface. The Package Explorer on the left shows the project structure: JavaProject > JRE System Library [JavaSE-13] > src > Assignment4 > encasptime.java. The main editor shows the source code of encasptime.java, which includes a Time class and a main method. The code is as follows:

```
33 System.out.println(Time1.toString());
34
35 System.out.println("Elapsed seonds in Time 1: " + Time1.getSeconds());
36
37 System.out.println("Enter Time 2 (Elapsed Time): ");
38
39 long mills = sc.nextInt();
40
41 time Time2 = new time(mills);
42
43 System.out.println(Time2);
44
45 System.out.println("Elapsed seconds in Time 2: " + Time2.getSeconds());
46
47 System.out.println("Time 1.compareTo(Time 2)? " + Time1.compareTo(Time2));
48
49 time T3 = Time1.Clone();
50
51 System.out.println("Time 3 is created as a clone of Time 1 Time1.compareTo(Time 3)? " + Time1.compareTo(T3));
52 System.out.print("<End Saple Run>");
53 sc.close();
54 }
55
56 }
57
```

The Outline view on the right shows the class structure: Assignment4 > encasptime > main(String[]) : void.

The Console view at the bottom shows the output of the program:

```
<terminated> encasptime [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Apr 8, 2022, 6:56:34 AM – 6:56:55 AM)
<sample run>
Enter time1 (hour minute second): 1 2 3
1 hours 2 minutes 3 seconds
Elapsed seonds in Time 1: 3723
Enter Time 2 (Elapsed Time):
193032
5 hours 37 minutes 12 seconds
Elapsed seconds in Time 2: 193032
Time 1.compareTo(Time 2)? -189309
Time 3 is created as a clone of Time 1 Time1.compareTo(Time 3)? 0
<End Saple Run>
```