
Practical - 03

=====

Student Name : Vinni Fengade
Roll No. : 67
Sem & Sec : 4 CSE [B]
Course Name : Object Oriented Programming (CSP256)
Date Compiled : 23-April-2022

=====

Problem Statements:

1. Write a program to demonstrate Inheritance of Classes.

A. Create a class Time with the data members as hours and minutes.

Add Functionality to add and subtract 2 time objects.

Test the time class is main().

B. "GreatClock" (A scientific research company) wants advanced time objects which will also provide functionalities of addition and subtraction of seconds and milliseconds.

How will you add this feature without changing the Time Class.

C. Create a class TimeZone which will add functionality to convert the time from one time zone to another time zone.

Note: Class Time zone uses the Time Object and uses the Add and Subtract methods of Time Class.

Code

File : Practical3.java

```
package com.mycompany.practical3;

public class Practical3 {

    public static void main(String[] args) {

//      A. Create a class Time with the data members as hours and minutes.
//      Add Functionality to add and subtract 2 time objects.

        Time t1 = new Time(5,24);
        Time t2 = new Time(6,25);
        Time t3 = new Time();
        t3.add(t1, t2);

        System.out.println("<-----Time-Clock----->");

        System.out.println("      Sum of Time : T1+T2");
        System.out.print("T1 -: ");
        t1.displaytime();
        System.out.print(" and   T2 -: ");
        t2.displaytime();
        System.out.print(" is  \n      T3 -: ");
        t3.displaytime();

        Time t4 = new Time();
        System.out.println("");
        t4.subtract(t1, t2);
        System.out.println("      Difference of Time : T1-T2");
        System.out.print("T1 -: ");
        t1.displaytime();
        System.out.print(" and   T2 -: ");
        t2.displaytime();
        System.out.print(" is  \n      T4 -: ");
        t4.displaytime();
```

```
//      B. "GreatClock" (A scientific research company) wants advanced time
//      objects which will also provide functionalities of addition and
//      subtraction of seconds and milliseconds.
//      How will you add this feature without changing the Time Class.

System.out.println("\n<-----Great-Clock----->");

GreatClock t11 = new GreatClock(5,24,22,05);
GreatClock t12 = new GreatClock(6,24,23,06);
GreatClock t13 = new GreatClock();
t13.add(t11, t12);

System.out.println("      Sum of Great Time : T11 + T12");
System.out.print("T11 -: ");
t11.displaytime();
System.out.print(" and T12 -: ");
t12.displaytime();
System.out.print(" is \n      T13 -: ");
t13.displaytime();

GreatClock t14 = new GreatClock();
t14.subtract(t11, t12);
System.out.println("\n      Difference of Time : T11-T12");
System.out.print("T11 -: ");
t11.displaytime();
System.out.print(" and T12 -: ");
t12.displaytime();
System.out.print(" is \n      T14 -: ");
t14.displaytime();
```

```
//      C. Create a class TimeZone which will add functionality to convert the
//      time from one time zone to another time zone.
      System.out.println("\n<-----Time-Zone----->");

      Time SrcTimeZone;
      SrcTimeZone = new Time(5,30);
      Time DestiTimeZone;
      DestiTimeZone = new Time(-7,0);
      Time SrcTime;
      SrcTime = new Time(21,52);

      TimeZone DestiTime;
      DestiTime = new TimeZone(SrcTimeZone, DestiTimeZone, SrcTime);
      System.out.print("Current Time in Source Time Zone :");
      SrcTime.displaytime();
      System.out.print("\nSource Time Zone :");
      SrcTimeZone.displaytime();
      System.out.print(" and Destination Time Zone :");
      DestiTimeZone.displaytime();
      System.out.print("\nCurrent time in Destination Time Zone is :");
      DestiTime.Z.displaytime();
      System.out.println();

  }
}
```

File : Time

```
package com.mycompany.practical3;
```

```
public class Time {
    int hrs,min;
    public Time() {
    }
    Time(int hrs,int min){
        this.hrs=hrs;
        this.min=min;
    }

    void add(Time t1,Time t2){
        int hr_sum=t1.hrs+t2.hrs;
        int min_sum=t1.min+t2.min;
        if(min_sum >=60){
            hr_sum++;
            min_sum=min_sum-60;
        }
        if(hr_sum>=24){
            hr_sum = hr_sum%24;
        }
        hrs=hr_sum;
        min=min_sum;
    }

    void subtract(Time t1,Time t2){
        int hr_sub= (24+t1.hrs-t2.hrs)%24;
        int min_sub= (60+t1.min-t2.min)%60;
        if((t1.min-t2.min) <0){
            hr_sub=hr_sub-1;
        }
        hrs=hr_sub;
        min=min_sub;
    }
}
```

```
void displaytime(){
    System.out.print(" "+ this.hrs +" : "+ this.min);
}
}
```

File : GreatClock

```
public class GreatClock extends Time{
    int seconds,milliseconds;

    public GreatClock() {
    }

    GreatClock(int hrs,int min,int seconds,int milliseconds){
        super(hrs,min);
        this.seconds=seconds;
        this.milliseconds=milliseconds;
    }

    void add(GreatClock t1,GreatClock t2){
        super.add(t1,t2);
        int seconds_sum=t1.seconds+t2.seconds;
        int milli_sum=t1.milliseconds+t2.milliseconds;
        if(milli_sum >=100){
            seconds_sum+=(milli_sum/100);
            milli_sum=milli_sum%100;
        }
        if(seconds_sum>=60){
            min++;
            seconds_sum = seconds_sum%60;
        }
        if(min>=60){
            hrs+=(min/60);
            min=min%60;
        }
        if(hrs>=24){
            hrs = hrs%24;
        }
        seconds=seconds_sum;
    }
}
```

```

        milliseconds=milli_sum;
        System.out.println(hrs +" : "+ min+" : "+ seconds+" : "+ milliseconds);
        System.out.println(seconds_sum+" : "+ milli_sum);
    }
    void subtract(GreatClock t1,GreatClock t2){
        super.subtract(t1,t2);
        int seconds_diff=t1.seconds-t2.seconds;
        int milli_diff=t1.milliseconds-t2.milliseconds;

        if(milli_diff <0){
            seconds_diff-=1;
            milli_diff=(milli_diff+100)%100;
        }
        if(seconds_diff<0){
            min--;
            seconds_diff = (60+seconds_diff)%60;
        }
        if(min<0){
            hrs-=1;
            min=(60+min)%60;
        }
        if(hrs<0){
            hrs = (24+hrs)%24;
        }
        seconds=seconds_diff;
        milliseconds=milli_diff;
    }
    void displaytime(){
        System.out.print(" "+ this.hrs +" : "+ this.min+" : "+ this.seconds+" : 
                                                                    "+ this.milliseconds);
    }
    void displaytime(int hr,int min){
        System.out.print(" "+ hr +" : "+ min);
    }
}

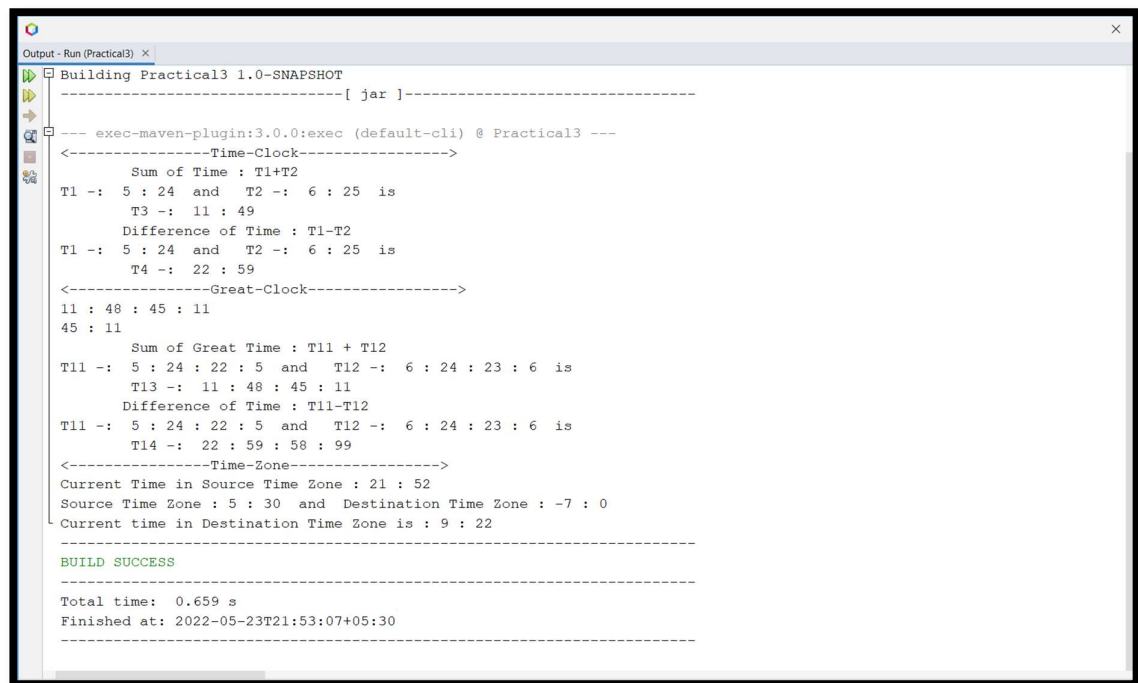
```

File : **TimeZone**

```
package com.mycompany.practical3;
```

```
public class TimeZone {  
    Time Z=new Time();  
    TimeZone(Time Src_z,Time Dest_z,Time Current_TSrc){  
        Z.subtract(Src_z, Dest_z);  
        Z.subtract(Current_TSrc,Z);  
    }  
}
```


Execution



```
Output - Run (Practical3) x
Building Practical3 1.0-SNAPSHOT
-----[ jar ]-----
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Practical3 ---
<-----Time-Clock----->
    Sum of Time : T1+T2
T1 -: 5 : 24 and T2 -: 6 : 25 is
    T3 -: 11 : 49
    Difference of Time : T1-T2
T1 -: 5 : 24 and T2 -: 6 : 25 is
    T4 -: 22 : 59
<-----Great-Clock----->
11 : 48 : 45 : 11
45 : 11
    Sum of Great Time : T11 + T12
T11 -: 5 : 24 : 22 : 5 and T12 -: 6 : 24 : 23 : 6 is
    T13 -: 11 : 48 : 45 : 11
    Difference of Time : T11-T12
T11 -: 5 : 24 : 22 : 5 and T12 -: 6 : 24 : 23 : 6 is
    T14 -: 22 : 59 : 58 : 99
<-----Time-Zone----->
Current Time in Source Time Zone : 21 : 52
Source Time Zone : 5 : 30 and Destination Time Zone : -7 : 0
Current time in Destination Time Zone is : 9 : 22
-----
BUILD SUCCESS
-----
Total time: 0.659 s
Finished at: 2022-05-23T21:53:07+05:30
-----
```