Computer Science 260

Homework 3: Immutable Classes

Due: Mon. Mar. 4 at the beginning of class

**Instructions**: Modify the following class so that it is immutable. Do this by identifying the mutators, and then rewrite these methods so that instead of changing the object that invoked the method, they create a new object that reflects the requested change. You only need to rewrite those methods that will have new implementations. *Do not change the public interface of the class.*

/\*\* A class to keep track of the time of day, to the nearest second.

\*/

public class TimeOfDay

{

public static final int HOURS\_PER\_DAY = 24;

public static final int MINUTES\_PER\_HOUR = 60;

public static final int SECONDS\_PER\_MINUTE = 60;

/\* The hours, minutes, and seconds that have

elapsed since midnight.

\*/

private int hours, minutes, seconds;

/\*\* Create a new object representing the time of day.

@param h the hour

@param m the minute

@param s the second

\*/

public TimeOfDay (int h, int m, int s)

{

hours = h;

minutes = m;

seconds = s;

}

/\*\* Get the hour represented by this object.

@return the hour

\*/

public int getHour() {return hours;}

/\*\* Get the minute represented by this object.

@return the minute

\*/

public int getMinute() {return minutes;}

/\*\* Get the second represented by this object.

@return the second

\*/

public int getsecond() {return seconds;}

/\*\* Advance the time by one second.

\*/

public TimeOfDay tick()

{

seconds++;

if (seconds == SECONDS\_PER\_MINUTE) {

seconds = 0;

minutes++;

if (minutes == MINUTES\_PER\_HOUR) {

minutes = 0;

hours++;

if (hours == HOURS\_PER\_DAY) {

hours = 0;

}

}

}

return this;

}

/\*\* Advance the time by a number of seconds

\* @param amount the number of seconds to advance

\*/

public TimeOfDay advance (int amount)

{

for (int i = 1; i <= amount; ++i) {

tick();

}

return this;

}

/\*\* Calculate the number of seconds elapsed since midnight.

\* @return the number of seconds elapsed since midnight

\*/

public int seconds\_elapsed ()

{

return (hours \* MINUTES\_PER HOUR + minutes)

\* SECONDS\_PER\_MINUTE + seconds;

}

/\*\* Calculate the number of hours elapsed since midnight.

\* @return the number of hours elapsed since midnight

\*/

public double hours\_elapsed ()

{

return (double) seconds\_elapsed ()

/ (double) (MINUTES\_PER\_HOUR \* SECONDS\_PER\_MINUTE);

/\*\* Advance the time of day by 12 hours.

\*/

public TimeOfDay flip()

{

hours = (hours + 12) % HOURS\_PER\_DAY;

return this;

}

/\*\* Determine if the time is in the morning or afternoon /

\* evening.

\* @return true if the time is before noon, and false otherwise

\*/

public boolean isAm ()

{

return hours < 12;

}

/\*\* Advance the time to the start of the next hour.

\*/

public TimeOfDay topOfTheHour()

{

second = minute = 0;

hour = (hour == HOURS\_PER\_DAY – 1)

? 0

: hour + 1;

return this;

}