Chapter 9 Lab
More Classes and Objects

Lab Objectives

- Use methods of the `Character` class and `String` class to process text
- Be able to use the `String.split` method and the `StringBuilder` class

Introduction

In this lab we ask the user to enter a time in military time (24 hours). The program will convert and display the equivalent conventional time (12 hour with AM or PM) for each entry if it is a valid military time. An error message will be printed to the console if the entry is not a valid military time.

Think about how you would convert any military time 00:00 to 23:59 into conventional time. Also think about what would be valid military times. To be a valid time, the data must have a specific form. First, it should have exactly 5 characters. Next, only digits are allowed in the first two and last two positions, and that a colon is always used in the middle position. Next, we need to ensure that we never have over 23 hours or 59 minutes. This will require us to separate the substrings containing the hours and minutes. When converting from military time to conventional time, we only have to worry about times that have hours greater than 12, and we do not need to do anything with the minutes at all. To convert, we will need to subtract 12, and put it back together with the colon and the minutes, and indicate that it is PM. Keep in mind that 00:00 in military time is 12:00 AM (midnight) and 12:00 in military time is 12:00 PM (noon).

We will need to use a variety of `Character` class and `String` class methods to validate the data and separate it in order to process it. We will also use a `Character` class method to allow the user to continue the program if desired.

The `String` class's `split` method will allow us to process the tokens in a text file in order to decode a secret message. We will use the first letter of every 5th token read in from a file to reveal the secret message.

Task #1 `Character` and `String` Class Methods

1. Copy the files *Time.java* (Code Listing 9.1) and *TimeDemo.java* (Code Listing 9.2) from the StudentCD or as directed by your instructor.
2. In the `Time.java` file, add conditions to the decision structure which validates the data. Conditions are needed that will:
    a. Check the length of the `String`
    b. Check the position of the colon
    c. Check that all other characters are digits

3. Add lines that will separate the `String` into two substrings containing hours and minutes. Convert these substrings to integers and save them into the instance variables.
4. In the `TimeDemo` class, add a condition to the loop that converts the user's answer to a capital letter prior to checking it.
5. Compile, debug, and run. Test out your program using the following valid input:
   a. 00:00
   b. 12:00
   c. 04:05
   d. 10:15
   e. 23:59
   f. 00:35

   And the following invalid input:
   a. 7:56
   b. 15:78
   c. 08:60
   d. 24:00
   e. 3e:33
   f. 1:111

Task #2 `String.split` and the `StringBuilder` Class

1. Copy the file *secret.txt* (Code Listing 9.3) from the Student CD or as directed by your instructor. This file is only one line long. It contains 2 sentences.
2. Write a `main` method that will read the file `secret.txt`, separate it into word tokens.
3. You should process the tokens by taking the first letter of every fifth word, starting with the first word in the file. Convert these letters to uppercase and append them to a `StringBuilder` object to form a word which will be printed to the console to display the secret message.

Code Listing 9.1 (`Time.java`)

```
/**
   Represents time in hours and minutes using
   the customary conventions.
*/

public class Time
{
   private int hours;          // Conventional hours
   private int minutes;        // Conventional minutes
   private boolean afternoon; // Flag for afternoon
```

```
/**
   Constructs a cutomary time (12 hours, am or pm)
   from a military time ##:##
   @param militaryTime Time in the military
          format ##:##
*/

public Time(String militaryTime)
{
   // Check to make sure something was entered
   if (militaryTime == null)
   {
      System.out.println(militaryTime +
                         " is not a " +
                         "valid miliary time." );
   }
   // Check to make sure there are 5 characters
   else if (// CONDITION TO CHECK LENGTH OF STRING)
   {
      System.out.println(militaryTime +
                         " is not a " +
                         "valid miliary time." );
   }
   else
   {
      // Check to make sure the colon is in
      // the correct spot
      if (//CONDITION TO CHECK COLON POSITION)
      {
         System.out.println(militaryTime +
                            " is not a " +
                            "valid miliary time." );
      }
      // Check to make sure all other characters
      // are digits
      else if (// CONDITION TO CHECK FOR DIGIT)
      {
         System.out.println(militaryTime +
                            " is not a " +
                            "valid miliary time." );
      }
```

```
else if (// CONDITION TO CHECK FOR DIGIT)
{
   System.out.println(militaryTime +
                      " is not a " +
                      "valid miliary time." );
}
else if (//CONDITION TO CHECK FOR DIGIT)
{
   System.out.println(militaryTime +
                      " is not a " +
                      "valid miliary time." );
}
else if (//CONDITION TO CHECK FOR DIGIT)
{
   System.out.println(militaryTime +
                      " is not a " +
                      "valid miliary time." );
}
else
{
   // SEPARATE THE STRING INTO THE HOURS
   // AND THE MINUTES, CONVERTING THEM TO
   // INTEGERS AND STORING INTO THE
   // INSTANCE VARIABLES

   // Validate hours and minutes are valid values
   if(hours > 23)
   {
      System.out.println(militaryTime +
                         " is not a " +
                         "valid miliary time." );
   }
   else if(minutes > 59)
   {
      System.out.println(militaryTime +
                         " is not a " +
                         "valid miliary time." );
   }
   // Convert military time to conventional time
   // for afternoon times
   else if (hours > 12)
   {
      hours = hours - 12;
      afternoon = true;
      System.out.println(this.toString());
   }
```

```
         // Account for midnight
         else if (hours == 0)
         {
            hours = 12;
            System.out.println(this.toString());
         }
         // Account for noon
         else if (hours == 12)
         {
            afternoon = true;
            System.out.println(this.toString());
         }
         // Morning times do not need converting
         else
         {
            System.out.println(this.toString());
         }
      }
   }
}

/**
   The toString method returns a conventional time.
   @return A conventional time with am or pm.
*/

public String toString()
{
   String am_pm;
   String zero = "";

   if (afternoon)
      am_pm = "PM";
   else
      am_pm = "AM";
   if (minutes < 10)
      zero = "0";

   return hours + ":" + zero + minutes + " " + am_pm;
}
}
```

Code Listing 9.2 (`TimeDemo.java`)

```
import java.util.Scanner;

/**
   This program demonstrates the Time class.
*/

public class TimeDemo
{
   public static void main(String[] args)
   {
      Scanner keyboard = new Scanner(System.in);
      char answer = 'Y';
      String enteredTime;
      String response;

      while (//CHECK ANSWER AFTER CONVERTING TO CAPITAL)
      {
         System.out.print("Enter a military time " +
                          "using the ##:## format: ");
         enteredTime = keyboard.nextLine();
         Time now = new Time (enteredTime);
         System.out.println("Do you want to enter " +
                            "another (Y/N)? ");
         response = keyboard.nextLine();
         answer = response.charAt(0);
      }
   }
}
```

Code Listing 9.3 (`secret.txt`)

```
January is the first month and December is the last.
Violet is a purple color as are lilac and plum.
```