```
1  namespace ExercismII
2  {
3      public class BookingBeauty
4      {
5          public enum DayOfWeek
6          {
7              Default,
8              Monday,
9              Tuesday,
10             Wednesday,
11             Thursday,
12             Friday,
13             Saturday,
14             Sunday
15         }
16
17         public enum Month
18         {
19             Default,
20             January,
21             February,
22             March,
23             April,
24             May,
25             June,
26             July,
27             August,
28             September,
29             October,
30             November,
31             December
32         }
33         public static DateTime Schedule(string
             appointmentDateDescription)
34         {
35
36             // Formato de fechas
37             // "7/25/2019 13:45:00"
38             // "July 25, 2019 13:45:00"
39             // "Thursday, July 25, 2019 13:45:00"
40
41             DateTime result;
42
43             int sign = 0;
44             for (int i = 0; i < appointmentDateDescription.Length; i++)
45             {
46                 if (appointmentDateDescription[i] == ',')
47                     sign++;
48             }
49
50             if (sign == 0)
51                 result = DateTimeOnlyNumber
                     (appointmentDateDescription);
```

```
52              else if (sign == 1)
53                  result = DateTimeWithMonth(appointmentDateDescription);
54              else
55                  result = DateTimeWithDayWeek
                      (appointmentDateDescription);
56
57              return result;
58          }
59
60          public static bool HasPassed(DateTime appointmentDate)
61          {
62              if (appointmentDate == null)
63                  return false;
64
65              DateTime dateNow = DateTime.Now;
66              DateTime date = appointmentDate;
67
68              if (date.Year < dateNow.Year)
69                  return true;
70
71              if (date.Month < dateNow.Month)
72                  return true;
73
74              if (date.Month == dateNow.Month && date.Day < dateNow.Day)
75                  return true;
76
77              if (date.Day == dateNow.Day && date.Hour < dateNow.Hour)
78                  return true;
79
80              if (date.Day == dateNow.Day && date.Hour == dateNow.Hour &&
                   date.Minute < dateNow.Minute)
81                  return true;
82
83              if (date.Day == dateNow.Day && date.Hour == dateNow.Hour &&
                   date.Minute == dateNow.Minute && date.Second <
                  dateNow.Second)
84                  return true;
85
86              return false;
87          }
88
89          public static bool IsAfternoonAppointment(DateTime
              appointmentDate)
90          {
91              if (appointmentDate == null)
92                  return false;
93              return appointmentDate.Hour >= 12 && appointmentDate.Hour <
                   18;
94          }
95
96          public static string Description(DateTime appointmentDate)
97          {
98              if (appointmentDate == null)
```

```
 99                 return "";
100
101             DateTime date = appointmentDate;
102             string meridian = "AM";
103             int month = date.Month;
104             int year = date.Year;
105             int day = date.Day;
106
107             int hour = date.Hour;
108             if (hour > 12)
109             {
110                 hour -= 12;
111                 meridian = "PM";
112             }
113             int minute = date.Minute;
114             string minuteFormat = minute.ToString("D2");
115             int second = date.Second;
116             string secondFormat = second.ToString("D2");
117
118             return $"You have an appointment on {month}/{day}/{year}
                  {hour}:{minuteFormat}:{secondFormat} {meridian}.";
119         }
120
121         public static DateTime AnniversaryDate()
122         {
123             return new DateTime(DateTime.Now.Year, 9, 15, 0, 0, 0);
124         }
125
126         // Metodos para Parsear un string en un formato 'DateTime'
127         public static DateTime DateTimeOnlyNumber(string
              appointmentDateDescription)
128         {
129             string appointment = appointmentDateDescription;
130             if (appointment[1] == '/')
131                 appointment = appointment.Insert(0, "0");
132
133             string month = appointment.Substring(0, 2);
134             string day = appointment.Substring(3, 2);
135             string year = appointment.Substring(6, 4);
136
137             string hour = appointment.Substring(11, 2);
138             string minute = appointment.Substring(14, 2);
139             string second = appointment.Substring(17, 2);
140
141             int monthInt = Int32.Parse(month);
142             int yearInt = Int32.Parse(year);
143             int dayInt = Int32.Parse(day);
144
145             int hourInt = Int32.Parse(hour);
146             int minuteInt = Int32.Parse(minute);
147             int secondInt = Int32.Parse(second);
148
149             DateTime date = new DateTime(yearInt, monthInt, dayInt,
```

```
                hourInt, minuteInt, secondInt);
150             return date;
151         }
152
153         public static DateTime DateTimeWithMonth(string
              appointmentDateDescription)
154         {
155             if (appointmentDateDescription == null)
156                 throw new Exception(appointmentDateDescription);
157
158             int i = 0;
159             string dateMonth = "";
160             string dateAppoint = appointmentDateDescription;
161             while (char.IsLetter(dateAppoint[i]) || char.IsNumber
                  (dateAppoint[i]))
162             {
163                 dateMonth += dateAppoint[i];
164                 i++;
165             }
166
167
168             string[] monthsName = Enum.GetNames<Month>();
169             int monthResult = 0;
170             for (int j = 0; j < 13; j++)
171             {
172                 if (dateMonth == monthsName[j])
173                     monthResult = j;
174             }
175
176             i++;
177
178             string dateDay = "";
179             while (char.IsLetter(dateAppoint[i]) || char.IsNumber
                  (dateAppoint[i]))
180             {
181                 dateDay += dateAppoint[i];
182                 i++;
183             }
184
185             i += 2;
186
187             string dateYear = "";
188             while (char.IsLetter(dateAppoint[i]) || char.IsNumber
                  (dateAppoint[i]))
189             {
190                 dateYear += dateAppoint[i];
191                 i++;
192             }
193
194             i++;
195
196
197             string hourString = "";
```

```
198             string minuteString = "";
199             string secondString = "";
200 
201             while (char.IsLetter(dateAppoint[i]) || char.IsNumber
                  (dateAppoint[i]))
202             {
203                 hourString += "" + dateAppoint[i];
204                 i++;
205             }
206             i++;
207 
208             while (char.IsLetter(dateAppoint[i]) || char.IsNumber
                  (dateAppoint[i]))
209             {
210                 minuteString += "" + dateAppoint[i];
211                 i++;
212             }
213             i++;
214 
215             while (i < dateAppoint.Length)
216             {
217                 secondString += "" + dateAppoint[i];
218                 i++;
219             }
220 
221             int year = int.Parse(dateYear);
222             int month = monthResult;
223             int day = int.Parse(dateDay);
224 
225             int hour = int.Parse(hourString);
226             int minute = int.Parse(minuteString);
227             int second = int.Parse(secondString);
228 
229             DateTime dateResult = new DateTime(year, month, day, hour,
                  minute, second);
230             return dateResult;
231         }
232 
233         public static DateTime DateTimeWithDayWeek(string
              appointmentDateDescription)
234         {
235             if (appointmentDateDescription == null)
236                 throw new Exception(appointmentDateDescription);
237 
238             int i = 0;
239             string dateWeek = "";
240             string dateAppoint = appointmentDateDescription;
241             while (char.IsLetter(dateAppoint[i]))
242             {
243                 dateWeek += dateAppoint[i];
244                 i++;
245             }
246 
```

```
247             string[] WeekName = Enum.GetNames<DayOfWeek>();
248             int WeekResult = 0;
249             for (int j = 1; j < 8; j++)
250             {
251                 if (dateWeek == WeekName[j])
252                     WeekResult = j;
253             }
254 
255             i += 2;
256 
257             string dateMonth = "";
258             while (char.IsLetter(dateAppoint[i]) || char.IsNumber
                  (dateAppoint[i]))
259             {
260                 dateMonth += dateAppoint[i];
261                 i++;
262             }
263 
264 
265             string[] monthsName = Enum.GetNames<Month>();
266             int monthResult = 0;
267             for (int j = 0; j < 13; j++)
268             {
269                 if (dateMonth == monthsName[j])
270                     monthResult = j;
271             }
272 
273             i++;
274 
275             string dateDay = "";
276             while (char.IsLetter(dateAppoint[i]) || char.IsNumber
                  (dateAppoint[i]))
277             {
278                 dateDay += dateAppoint[i];
279                 i++;
280             }
281 
282             i += 2;
283 
284             string dateYear = "";
285             while (char.IsLetter(dateAppoint[i]) || char.IsNumber
                  (dateAppoint[i]))
286             {
287                 dateYear += dateAppoint[i];
288                 i++;
289             }
290 
291             i++;
292 
293             string hourString = "";
294             string minuteString = "";
295             string secondString = "";
296 
```

```
297             while (char.IsLetter(dateAppoint[i]) || char.IsNumber
                  (dateAppoint[i]))
298             {
299                 hourString += "" + dateAppoint[i];
300                 i++;
301             }
302             i++;
303 
304             while (char.IsLetter(dateAppoint[i]) || char.IsNumber
                  (dateAppoint[i]))
305             {
306                 minuteString += "" + dateAppoint[i];
307                 i++;
308             }
309             i++;
310 
311             while (i < dateAppoint.Length)
312             {
313                 secondString += "" + dateAppoint[i];
314                 i++;
315             }
316 
317             int year = int.Parse(dateYear);
318             int month = monthResult;
319             int day = int.Parse(dateDay);
320 
321             int hour = int.Parse(hourString);
322             int minute = int.Parse(minuteString);
323             int second = int.Parse(secondString);
324 
325             DateTime dateResult = new DateTime(year, month, day, hour,
                  minute, second);
326             return dateResult;
327         }
328     }
329 }
330 
```