```
namespace ExercismII
{
    public class BookingBeauty
    {
        public enum DayOfWeek
        {
            Default,
            Monday,
            Tuesday,
            Wednesday,
            Thursday,
            Friday,
            Saturday,
            Sunday
        }

        public enum Month
        {
            Default,
            January,
            February,
            March,
            April,
            May,
            June,
            July,
            August,
            September,
            October,
            November,
            December
        }
        public static DateTime Schedule(string appointmentDateDescription)
        {

            // Formato de fechas
            // "7/25/2019 13:45:00"
            // "July 25, 2019 13:45:00"
            // "Thursday, July 25, 2019 13:45:00"

            DateTime result;

            int sign = 0;
            for (int i = 0; i < appointmentDateDescription.Length; i++)
            {
                if (appointmentDateDescription[i] == ',')
                    sign++;
            }

            if (sign == 0)
                result = DateTimeOnlyNumber(appointmentDateDescription);
            else if (sign == 1)
                result = DateTimeWithMonth(appointmentDateDescription);
            else
                result = DateTimeWithDayWeek(appointmentDateDescription);

            return result;
        }

        public static bool HasPassed(DateTime appointmentDate)
        {
            if (appointmentDate == null)
                return false;

            DateTime dateNow = DateTime.Now;
            DateTime date = appointmentDate;

            if (date.Year < dateNow.Year)
                return true;

            if (date.Month < dateNow.Month)
                return true;

            if (date.Month == dateNow.Month && date.Day < dateNow.Day)
                return true;

            if (date.Day == dateNow.Day && date.Hour < dateNow.Hour)
                return true;

            if (date.Day == dateNow.Day && date.Hour == dateNow.Hour && date.Minute < dateNow.Minute)
                return true;

            if (date.Day == dateNow.Day && date.Hour == dateNow.Hour && date.Minute == dateNow.Minute && date.Second < dateNow.Second)
                return true;

            return false;
        }

        public static bool IsAfternoonAppointment(DateTime appointmentDate)
        {
            if (appointmentDate == null)
                return false;
            return appointmentDate.Hour >= 12 && appointmentDate.Hour < 18;
        }

        public static string Description(DateTime appointmentDate)
        {
            if (appointmentDate == null)
```

```
                return "";

            DateTime date = appointmentDate;
            string meridian = "AM";
            int month = date.Month;
            int year = date.Year;
            int day = date.Day;

            int hour = date.Hour;
            if (hour > 12)
            {
                hour -= 12;
                meridian = "PM";
            }
            int minute = date.Minute;
            string minuteFormat = minute.ToString("D2");
            int second = date.Second;
            string secondFormat = second.ToString("D2");

            return $"You have an appointment on {month}/{day}/{year} {hour}:{minuteFormat}:{secondFormat} {meridian}.";
        }

        public static DateTime AnniversaryDate()
        {
            return new DateTime(DateTime.Now.Year, 9, 15, 0, 0, 0);
        }

        // Metodos para Parsear un string en un formato 'DateTime'
        public static DateTime DateTimeOnlyNumber(string appointmentDateDescription)
        {
            string appointment = appointmentDateDescription;
            if (appointment[1] == '/')
                appointment = appointment.Insert(0, "0");

            string month = appointment.Substring(0, 2);
            string day = appointment.Substring(3, 2);
            string year = appointment.Substring(6, 4);

            string hour = appointment.Substring(11, 2);
            string minute = appointment.Substring(14, 2);
            string second = appointment.Substring(17, 2);

            int monthInt = Int32.Parse(month);
            int yearInt = Int32.Parse(year);
            int dayInt = Int32.Parse(day);

            int hourInt = Int32.Parse(hour);
            int minuteInt = Int32.Parse(minute);
            int secondInt = Int32.Parse(second);

            DateTime date = new DateTime(yearInt, monthInt, dayInt,
              hourInt, minuteInt, secondInt);
            return date;
        }

        public static DateTime DateTimeWithMonth(string appointmentDateDescription)
        {
            if (appointmentDateDescription == null)
                throw new Exception(appointmentDateDescription);

            int i = 0;
            string dateMonth = "";
            string dateAppoint = appointmentDateDescription;
            while (char.IsLetter(dateAppoint[i]) || char.IsNumber(dateAppoint[i]))
            {
                dateMonth += dateAppoint[i];
                i++;
            }


            string[] monthsName = Enum.GetNames<Month>();
            int monthResult = 0;
            for (int j = 0; j < 13; j++)
            {
                if (dateMonth == monthsName[j])
                    monthResult = j;
            }

            i++;

            string dateDay = "";
            while (char.IsLetter(dateAppoint[i]) || char.IsNumber(dateAppoint[i]))
            {
                dateDay += dateAppoint[i];
                i++;
            }

            i += 2;

            string dateYear = "";
            while (char.IsLetter(dateAppoint[i]) || char.IsNumber(dateAppoint[i]))
            {
                dateYear += dateAppoint[i];
                i++;
            }

            i++;


            string hourString = "";
            string minuteString = "";
            string secondString = "";

            while (char.IsLetter(dateAppoint[i]) || char.IsNumber(dateAppoint[i]))
            {
                hourString += "" + dateAppoint[i];
                i++;
            }
            i++;

            while (char.IsLetter(dateAppoint[i]) || char.IsNumber(dateAppoint[i]))
            {
                minuteString += "" + dateAppoint[i];
                i++;
            }
            i++;

            while (i < dateAppoint.Length)
            {
                secondString += "" + dateAppoint[i];
                i++;
            }

            int year = int.Parse(dateYear);
            int month = monthResult;
            int day = int.Parse(dateDay);

            int hour = int.Parse(hourString);
            int minute = int.Parse(minuteString);
            int second = int.Parse(secondString);

            DateTime dateResult = new DateTime(year, month, day, hour, minute, second);
            return dateResult;
        }

        public static DateTime DateTimeWithDayWeek(string appointmentDateDescription)
        {
            if (appointmentDateDescription == null)
                throw new Exception(appointmentDateDescription);

            int i = 0;
            string dateWeek = "";
            string dateAppoint = appointmentDateDescription;
            while (char.IsLetter(dateAppoint[i]))
            {
                dateWeek += dateAppoint[i];
                i++;
            }

            string[] WeekName = Enum.GetNames<DayOfWeek>();
            int WeekResult = 0;
            for (int j = 1; j < 8; j++)
            {
                if (dateWeek == WeekName[j])
                    WeekResult = j;
            }

            i += 2;

            string dateMonth = "";
            while (char.IsLetter(dateAppoint[i]) || char.IsNumber(dateAppoint[i]))
            {
                dateMonth += dateAppoint[i];
                i++;
            }


            string[] monthsName = Enum.GetNames<Month>();
            int monthResult = 0;
            for (int j = 0; j < 13; j++)
            {
                if (dateMonth == monthsName[j])
                    monthResult = j;
            }

            i++;

            string dateDay = "";
            while (char.IsLetter(dateAppoint[i]) || char.IsNumber(dateAppoint[i]))
            {
                dateDay += dateAppoint[i];
                i++;
            }

            i += 2;

            string dateYear = "";
            while (char.IsLetter(dateAppoint[i]) || char.IsNumber(dateAppoint[i]))
            {
                dateYear += dateAppoint[i];
                i++;
            }

            i++;

            string hourString = "";
            string minuteString = "";
            string secondString = "";

            while (char.IsLetter(dateAppoint[i]) || char.IsNumber(dateAppoint[i]))
            {
                hourString += "" + dateAppoint[i];
                i++;
            }
            i++;

            while (char.IsLetter(dateAppoint[i]) || char.IsNumber(dateAppoint[i]))
            {
                minuteString += "" + dateAppoint[i];
                i++;
            }
            i++;

            while (i < dateAppoint.Length)
            {
                secondString += "" + dateAppoint[i];
                i++;
            }

            int year = int.Parse(dateYear);
            int month = monthResult;
            int day = int.Parse(dateDay);

            int hour = int.Parse(hourString);
            int minute = int.Parse(minuteString);
            int second = int.Parse(secondString);

            DateTime dateResult = new DateTime(year, month, day, hour, minute, second);
            return dateResult;
        }
    }
}

```