

Да се разпише клас **Day** със свойства (след създаване не могат да се променят):

- **Date:** номер на дена в месеца (между 1 и 31)
- **Temperature:** реално число – между -50 и 50
- **Weather:** едно от четирите - sunny, cloudy, windy, rainy (да се ползва enum)

Да се предефинира метода ToString за да показва информацията за датата във вида:

{date}: {weather}, {temperature} °C

За символа на градуса може да се ползва Unicode \u00B0

Да се разпише клас **Month** със свойства:

- **Name:** името на месеца (да се ползва Enum)
- **NumberOfDays:** според името на месеца
- **Days:** колекция от елементи от тип Day (изберете подходящ вид)

Да се съставят следните методи:

- **AddDay** – за добавяне на ден, ако е валиден
- **RemoveDay** – премахва деня по зададена дата ако го има в колекцията
- **AvgTemperature** – извежда средната температура в случай че са попълнени всички дни или вади грешка “Missing Data!”
- **MinTemperature** – извежда най-ниската температура за месеца в случай че са попълнени всички дни или вади грешка “Missing Data!”
- **MaxTemperature** – извежда най-високата температура за месеца в случай че са попълнени всички дни или вади грешка “Missing Data!”
- **MonthType** – връща sunny, cloudy, windy, rainy в случай, че повече от половината от дните за месеца са били от този тип или mixed
- **DaysAbove(температура)** – връща броя на дните с температура над посочената
- **DaysBelow(температура)** – връща броя на дните с температура под посочената
- **предефинирана ToString** – връща информация във формат:
 - sunny days: { *изчисления брой* }
 - windy days: { *изчисления брой* }
 - cloudy days: { *изчисления брой* }
 - rainy days: { *изчисления брой* }
 - min temperature: { *изчислената минимална температура* }
 - max temperature: { *изчислената максимална температура* }

Да се подготви подходящ код за тестване на горните функции.