Kodtest Norion bank av Björn Nilsson

I denna uppgiften har jag utgått från att koden jag har blivit tilldelad är en del av en större struktur, och därmed försökt jobba med metodsignaturerna jag har blivit given.

Ändringar och förbättringar:

Tollcalculators constructor tar nu en int för att initiera nuget paketet PublicHoliday, detta för att minimera möjligtvis dyra API calls. Du matar in en int som representerar året du vill kalkylera på, detta bygger en lista med alla årets lediga dagar,helgdagar och så vidare. Informationen finns tillgänglig i _Holidays privata lista inuti TollCalculator

Till att börja med har jag tagit bort enumen som representerade skattefria fordon, och ersatt den med en List<Type> TollFreeVehicles.

Jag har skapat en en const som representerar en MaxTollFee , den maximala summan du kan få på en dag.

Härnäst kommer List<DayOfWeek> FreeWeekDays, som är en lista som representerar de fria dagarna på en vecka som man enkelt kan addera om ändringar i framtiden skulle ske, jag valde detta då det är en simpel utbyggbar metodik för att hantera detta.

TimeAndFee
Denna klass representerar en starttid, sluttid och ett pris för en tidsintervall och kostnad som används för att bygga listor med intervalltider för att beräkna kostnader. Den har tillhörande privata fält relaterade för kalkyleringen av detta, och även en intern metod “IsInTollTime” vilket gör “Lookup” simpelt och snabbt.

Sedermera kommer List<TimeAndFee> vilket representerar en lista av klassen TimeAndFee vilken representerar våra tider då tull ska tas ut och vilka intervaller den ska tas, och avgiften relaterad till denna tidsintervall.

Vehicle: Detta interface har jag döpt om till IVehicle, eftersom det följer rekommenderade konventioner, trots att det möjligtivs kan ha negativ påverkan på kodbasen. Men interfaces bör starta med I för att inte man ska blanda ihop dom med klasser till exempel.

GetTollFee(IVehicle vehicle, DateTime[] dates)
Denna klass är där merparten av förändringarna har skett. Jag resonerade som så att ni medvetet ville att denna endast skulle hantera ett enda datum baserat på era kommentarer i koden, dock ogillade jag att ni kallade en annan overload av samma namn inuti denna, och har refaktorerat detta. Denna metod kommer nu att kasta felmeddelande om du förser den med ett null vehicle eller dates array. sen kommer den kalla IsTollFreeVehicle/IsTollFreeDate och göra en early exit genom att returnera 0 om någon av dessa bool metoder returnerar true. Vi gör sen en verifieringskoll och kollar så vi inte blivit passade en array med multipla datum då jag antog baserat på hur strukturen var gjord att detta inte var avsett att hanteras av denna metoden eftersom den returnerar en int. om så är fallet kastar vi ett argument exception. vi sorterar sedan arrayen eftersom detta är viktigt för metodens funktion.

vi deklarerar sedan en lista med “GraceTimes” vilket är våran Lista med listor med tim-intervaller att kalkylera avgifter på eftersom multipla passager inom en timme är gratis. vi deklarerar sen en lista med gracePeriod som är våra individuella timlistor. Vi loopar på arrayen och hittar tider som är inom timespan 60 minuter från våran tidigare deklarerade StartTime vilket är arrayens första tid. om den är inom intervallen så adderar vi tiden till våran GracePeriod, annars lägger vi till våran våran graceperiod i graceTimes listan, återinitierar graceperiod , sätter starttime till loopens interna date variabel, och lägger till det nya startdatumet som start för den nya graceperiod. om vi har nått slutet på dates, adderar vi graceperiod till gracetimes. loopen är i detta fall avslutad. Vi loopar sedermera på vår lista av listor och använder timesandfees för att kolla om tiden är avgiftsbelagd, och vilken tid som har högst avgift, den tiden med högst avgift inom en intervall är den enda som kommer adderas till våran totala kostnad. om totalkostnad är över maxkonstanten för daglig avgift, sätter vi värdet av totalfee till MaxTollFee, vi returnerar sen vad vi har kalkylerat på TotalFee till callsite.

GetTollfee(DateTime, Ivehicle vehicle) Denna metod används inte längre men är simplifierad med användandet av TimeAndFee och ett simpelt linq statement.

IsTollFreeVehicle använder ett simpelt contains statement på TollFreeVehicles och returnerar bool värdet med resultatet.

List<int> FreeMonths är för att lagra gratis månader i systemet, i detta fallet månaden Juli som deklarerat i krav specifikationen, med simpel utbyggnad om så behövs.

IsTollFreeDate(DateTime date)

Denna metod gör simpla .Contains looups på FreeWeekDays, _Holidays och även kollar om imorgon är en ledig dag, isåfall är även idag en fri tulldag.