Parking System

1. Въведение

Тази програма е система за управление на паркинг. Целта на приложението е да се създаде автоматизирана система за управление на паркинг.

2.Основна функционалност

* Избор на място

Потребителите могат да видят списък, със паркоместата, и техният текущ статус - свободни/заети, от кого са заети, както и кога ще бъдат освободени заетите паркоместа.

* Запазване

Потребителите заявяват вариант за запазване, както и продължителност на резервацията.

* Варианти за запазване – Потребителите могат да запазят свободно паркомясто за определен период от време спрямо 3 варианта :
  + Предварително запазване: До 1 седмица преди началото на резервацията, потребителите могат да запазят желано свободно паркомясто за период от 1 ден до 1 седмица.
  + Запазване на момента: Потребителите могат да запазят желано паркомясто от 1 час до цял ден, а ако запазват през уикенда могат да запазват за период от 1 час до края на уикенда (събота и неделя).
  + Абонамент: Потребителите могат да заявят месечен абонамент ефективен от момента на запазване с продължителност 30 дена. При изтичане на абонамента паркомястото бива освободено
* Освобождаване на паркомясто - Потребителите имат опция за предварително напускане на паркинга (без абонамент), като се възстановяват 70% от оставащата сума.

3. Структура и имплементация

За създаването на програмата ни беше нужно да създадем няколко класа – class Transaction; class ParkingSys; class Parking; class TransactionHelper; class ParkingTicket; class OnTheGoParkingTicket; class SubscriptionParkingTicket; class InAdvanceParkingTicket.

**Class Parking**

Класът Parking е физическата репрезентация на паркинга. В него има property ParkingSpots (string[]), в което се съдържа всяко едно паркинг място. В него също така има и статичен метод, чрез който даваме стойности на това property.

**Class ParkingTicket**

Класът ParkingTicket се използва за да се представи всеки паркинг билет. Този клас е абстрактен.Той има няколко property-та – за кога започва резервацията, направена от купувач, за кога резервацията ще свърши, колко е нужно купувачът да плати за 1 час, колко струва изцяло билетът, името на паркинг мястото, където колата на купувача ще бъде, както и регистрацията на колата на купувача.

Този клас има няколко метода – CalculateTicketCost (чрез него се открива колко е пълната цена на паркинг билета); IsSpotFreeAndContained (чрез него се проверява дали мястото, където купувача иска да остави колата си, в момента е запазено; PrintOutTicketInfo (извежда цялата информация, свързана с паркинг билета)

**Classes OnTheGoParkingTicket, InAdvanceParkingTicket, SubscriptionParkingTicket**

Тези три класа наследяват от класът ParkingTicket. Трите класа не се различават по много. Различават се само по тяхната цена и продължителност. Те имат метод, чрез който потребителя може да създаде свой паркинг билет.

**Class Transaction**

Класът Transaction отговаря за всички транзакции. Точно чрез него може да се направят всички резервации на паркинг места, както и ранното освобождаване на паркинг местата.

В този клас има няколко property-та : ParkingTicket (в него се съдържа информация за паркинг билета, свързан с тази транзакция); expiredTransactions (съдържа информация за всички вече свършили транзакции); activeTransactions (съдържа информация за всички активни транзакции); comingUpTransactions (съдържа информация за всички предстоящи транзакции);

В този клас има статичен конструктор, чрез който се проверява дали всички транзакции са все още активни или е време да бъдат активирани.

В този клас има няколко метода : CheckTransactions\_OnDaysSwitched (При сменяне на ден се проверяват всички транзакции); TryReserveParkingSpot (Този метод се използва, когато потребителят би искал да запази паркинг място); TryFreeUpParkingSpaceEarlier (Чрез този метод може да се освободи паркинг място преди неговата крайна дата); ShowAllTransactions (Показва всички транзакции); CheckIfSpotIsNOTReservedAheadOrNow (Този метод проверява дали вече има направена резервация на съответното паркинг място, когато паркинг билета, който искаме да създадем, е активен)

**Class TransactionHelper**

Този клас е създаден с цел да “помага“ на Transaction класа. Чрез този клас може от текстови файлове да се взема информация за вече създадени транзакции, както и информация за текущата датата в програмата (това не е днешната дата, а тази в програмата). Той също така може и да записва съответно създадените транзакции в текстови файлове, за да може да бъдат запазени за бъдещо използване, след затваряне на програмата.

**Class ParkingSys**

В този клас се изпълняват всички функционалности на класовете, споменати отгоре. Чрез този клас се извикват методите за резервиране на паркинг места, освобождаване на паркинг места, преминаване на следващия ден и така нататък.

В този клас се намира методът TimeLoop. Този метод бива извикан в Program.cs и чрез него се изпълнява цялата функционалност на програмата. В този метод има няколко команди, които потребителят може да изпълнява, в зависимост от това дали е администратор или обикновен потребител :

* park-spot-info (Показва информация за всички паркови места)
* show-transactions (Показва всички транзакции)
* skip-day (Пропуска сегашния ден без да прави промени)
* exit (Изход от програмата)
* reserve-spot (Резервира парково място)
* free-spot (Освобождава парково място, преди да дойде крайната дата)

Също така в този клас се намира и методът SwitchDays, който прави програмата по-интерактивна. Чрез него веднага се преминава на следващия ден, като се праща с това “нотификация“ на всички транзакции, за да проверят дали е време да се освободи паркинг мястото, запазено от паркинг билета на съответната транзакцията.