1. При старте наша программа загружает конфиг файл и сохраняет значения в статическом классе AppSettings. Класс AppSettings содержит следующие данные

* FreePeriodInSeconds
* DailyReportPeriodInSeconds
* PricePerSeconds
* SendEndFreePeriodEmailNotifications
* SendEndFreePeriodSmsNotifications

Создается Parking Manager и класс Parking который либо достает данные из json файла или создает новый инстанс класса.

1. Так же при старте запускается бесконечный цикл, который все время ожидает команды от пользователя. Команд должно быть 4

- (add или ADD или aDd) – добавить машину на парковку

- (remove или Remove или ReMove) – удалить машину с парковки

- (getAll или GeTaLL) – получить список всех машин на парковке

- (q или Q или Quit) – завершить программу

Кроме этого, при старте запускается таймер, который через DailyReportPeriodInSeconds должен создать отчет сколько машин сегодня было на парковке, сколько в среднем они находились на парковке и сколько денег сегодня заплатили клиенты. Данный отчет должен сохраняться в папку DailyReports (если нет такой папки – нужно создать) и каждое имя отчета должно иметь формат { Year\_Month\_Day\_Hours\_Minutes\_Seconds\_Milliseconds}

1. При выполнении команды **add** – машина добавляется на парковку ей присваивается уникальный id и устанавливается время, когда она заехала на парковку.
2. После добавления машины на парковку запускается таймер, который через FreePeriodInSeconds отправляет нотификации по Email и SMS пользователю об окончании бесплатного периода.
3. Для реализации нотификаций нужно сделать интерфейс INotification с методом Notify который будет выполнять отправку (здесь для каждой реализации нужно учитывать есть ли у пользователя Phone, Email и установлены ли SendEndFreePeriodEmailNotifications и SendEndFreePeriodSmsNotifications в true)
4. При выполнении команды **remove** – машина удаляет из парковки. Перед тем как удалить машину мы должны проверить закончился ли у пользователя бесплатный период. Если период еще не закончился – просто удаляем машину с парковки и добавляем ее в коллекцию для дневного отчета. Если бесплатный период закончился мы должны вывести на экран сообщение что вы должны оплатить N сумму, которую высчитываем из времени пребывания машины на парковке и умножаем на PricePerSecond.
5. У клиента есть два пути оплаты. 1 – наличными 2- картой. Т.е. ваша консоль должны ожидать два варианта ввода 1 – cash 2 - card.
6. Для реализации функциональности оплаты должен быть создать интерфейс IPayment с методом Pay. Далее нужно реализовать два класса CashPayment и CardPayment которые реализуют интерфейс IPayment.
7. При выполнении команды **getall** – нужно вывести id и марку машин, которые в текущий момент есть на парковке.
8. При выполнении команды **q** – завершить программу. Перед завершением все машины, которые были на паркинге сериализуются в json файл для того чтобы восстановить состояние при следующем запуске программы.
9. Отчет, который запускался в самом начале через период DailyReportPeriodInSeconds после выполнения должен быть заново создан таймер с DailyReportPeriodInSeconds периодом и все данные, которые мы собирали за этот период должны быть очищены из коллекции.
10. Отчет сбрасывается на диск и содержит следующую информацию

- Информация о количестве машин за день

- Среднее количество времени в секундах которое провели машины на парковке

- Данные о клиентах, которые оплатили {FirstName, LastName, Email, Phone)

- Средняя сумма оплаты

- Максимальная сумма оплаты и информация о клиенте

- Количество оплат по карте и наличными

Классы, которые у нас будут учувствовать в программе

1. AppSettings – настройки нашего приложения
2. ParkedCar – представляет машину которая находится на парковке
3. Parking – описывает саму парковку. Должен содержать
4. ParkingManager – содержит ссылки на Parking и есть метод который возвращает IPayment для оплаты. IPayment выбирается на основании того какой тип оплаты выбрал пользователь.
5. DailyReport – статический класс который собирает информацию для отчета. Обнуляется после создания отчета.
6. IPayment, CashPayment, CardPayment – интерфейс и классы для оплаты парковки.
7. INotification, EmailNotification, SmsNotification – интерфейс и классы для отправки нотификаций.