Napisz program, który będzie służył do zarządzania przez właściciela siecią serwisów samochodowych. Właściciel może posiadać wiele serwisów jak i magazynów, które znajdują się w obrębie serwisów, w których istnieje dowolna (skończona) liczba powierzchni magazynowych do wynajęcia. Dodatkowo zawsze pewien procent magazynu (założony podczas inicjalizacji obiektu) stanowi magazyn serwisu, w który składowane są pomniejsze części oraz powierzchnia magazynowa przeznaczona na składowanie opon letnich/zimowych klientów poza sezonem za stosowną opłatą.

Każde pomieszczenie magazynowe może posiadać dowolną liczbę osób uprawnionych do jego otwarcia i wyjmowania/składowania przedmiotów. Informacje o takich osobach należy przechowywać w postaci obiektów klasy ***Person***), jednak należy pamiętać, że zawsze pierwsza osoba, która rozpoczyna najem pomieszczenia jest osobą odpowiedzialną za opłaty dot. wynajmu.

W ramach najmu powierzchni serwisu jest możliwość również dodatkowego najmu miejsca parkingowego na czas po naprawie (w przypadku braku możliwości terminowego odbioru samochodu po naprawie, jednak czas ten nie może być dłuższy niż 14 dni.

Każde pomieszczenie magazynowe jak i miejsce parkingowe posiadają informację o swojej powierzchni użytkowej jak i unikalny numer identyfikacyjny, dzięki któremu możemy jednoznacznie zdefiniować obiekt symbolizujący pomieszczenie magazynowe lub miejsce parkingowe.

Najemca może dowolnie nadawać i odbierać uprawnienia do pomieszczenia magazynowego, jak i również każda uprawniona osoba może dowolnie może wkładać i wyjmować przedmioty. Każda osoba może mieć wynajętych wiele pomieszczeń magazynowych, jak i miejsc na przechowanie pojazdu po naprawie, o ile sumaryczny koszt wynajmu nie będzie przekraczał 1250 zł (ceny najmu są dowolne).

Należy zapewnić, aby identyfikatory pomieszczeń i miejsc parkingowych były unikalne oraz tworzone automatycznie podczas tworzenia obiektu typu ***ConsumerWarehouse***, ***ServiceWarehouse*** i ***ParkingSpace***. Rozmiar powierzchni użytkowej w przypadku wspomnianych typów należy dostarczać podczas tworzenia obiektu pomieszczenia.

Serwis posiada również określoną liczbę wyspecjalizowanych miejsc naprawczych (***CarServiceSpot***) jak i co najmniej jedno samodzielne miejsce serwisowe (***IndependentCarServiceSpot***), które każda osoba może wynająć i wykonać samodzielnie prace serwisowe przy pojeździe.

Możliwe są dwa sposoby wskazania objętości pomieszczenia:
- poprzez podanie objętości w metrach sześciennych
- poprzez podanie rozmiarów pomieszczenia w postaci długości, szerokości i wysokości pomieszczenia (dla uproszczenia zakładamy, że pomieszczenie jest prostopadłościanem. Przy tym podejściu powierzchnia użytkowa pomieszczenia zostaje wyliczona podczas tworzenia obiektu na podstawie przekazanych wartości).

Pomieszczenie dla określonego najemcy posiada również datę rozpoczęcia najmu oraz datę zakończenia najmu. Jeśli data zakończenia najmu się przedawniła, to zostaje wystosowane pismo (obiekt typu ***TenantAlert***), które zostaje zapisane do obiektu klasy ***Person*** definiującego konkretnego najemcę.

Należy również zaimplementować mechanizm upływającego czasu za pośrednictwem wątków. Wątek powinien co 5 sekund przesuwać datę o 1 dzień do przodu, symulując upływ czasu. Równolegle powinny być co 10 sekund sprawdzane kwestie wynajmu, czy wszystkie pomieszczenia nadal są w trakcie najmu, czy może wynajem pomieszczenia już ustał.

Jeśli najem poza ustalonym w umowie terminem zostanie opłacony przez najemcę, pismo dot. zadłużenia zostaje usunięte z akt osoby. Jeśli najemca nie odnowi najmu w przeciągu 30 dni, należy wyczyścić pomieszczenie, którego najem się zakończyło, a pismo pozostaje w aktach.

W przypadku miejsca parkingowego, w pierwszej kolejności sprawdzamy czy znajduje się na miejscu parkingowym pojazd. Jeśli tak, to w związku z przyśpieszoną decyzją administracyjną, zostaje od odholowany na specjalnie do tego wyznaczony płatny parking strzeżony (z punktu widzenia aplikacji po prostu usuwamy pojazd ze wskazanego miejsca parkingowego) i ostaje wystosowane pismo (obiekt typu ***TenantAlert***), które zostaje zapisane do obiektu klasy ***Person*** definiującego konkretnego najemcę.

Osoba poza polami potrzebne do realizacji zadania, musi posiadać takie dane jak imię, nazwisko, pesel, adres zamieszkania, datę urodzenia oraz datę pierwszego wykonanego przez nią najmu pomieszczenia. W przypadku, gdy osoba nie wynajęła nigdy żadnego pomieszczenia w sytuacji próby pobrania tej daty powinien zostać rzucony wyjątek ***NeverRentException***.

Jeśli najem będzie chciała rozpocząć osoba z więcej niż trzema zadłużeniami (co najmniej 3 obiekty typu ***Info***) na przestrzeni najmów na tle całego osiedla, rzucony powinien zostać wyjątek ***ProblematicTenantException*** z komunikatem – *„Osoba X posiadała już najem pomieszczeń: lista_pomieszczeń - wysokość_zadłużenia”*, gdzie X to imię i nazwisko danej osoby, *lista_pomieszczeń* definiuje wynajmowane pomieszczenia, zaś *wysokość_zadłużenia* definiuje sumaryczną wysokość zadłużenia za to pomieszczenie.

Każdy z przedmiotów posiada zapisaną informację o nazwie oraz polu powierzchni jaką zajmuje. Powierzchnia zajmowana przez przedmiot może być dostarczona na dwa sposoby tak jak w przypadku pomieszczenia.

W przypadku wkładania każdego przedmiotu do pomieszczenia musimy się upewnić, że pomieszczenie jest w stanie pomieścić ten obiekt. Jeśli tak nie jest, zostaje rzucony wyjątek ***TooManyThingsException*** z komunikatem *„Remove some old items to insert a new item”*.

W ramach serwisu osoba oczywiście może również zgłaszać potrzebę serwisową dot. swojego pojazdu. Dla uproszczenia zakładamy, ze zgłoszenie serwisowe jest jednoznaczne ze wstawieniem pojazdu do wyznaczonego miejsca serwisowego (w zależności od sposobu naprawy pojazdu - naprawa przez mechanika lub samodzielna, których ceny są dowolne). W przypadku, gdy wszystkie miejsca serwisowe danego typu są zajęte, klient zostaje wpisany na listę oczekujących, samochód odstawiony na miejsce parkingowe (za darmo) i po uwolnieniu się miejsca serwisowego automatycznie samochód zostanie przyjęty do serwisu.

Każda naprawa serwisowa ma określony czas naprawy, który jest losowany z zakresu 1-5 dni (dla uproszczenia nie rozróżniamy dni roboczych i dni wolnych).

Pojazdy są podzielone na rozróżniające je typy:

- samochód terenowy,
- samochód miejski,
- motocykl,
- amfibia

Każdy z pojazdów poza nazwą oraz powierzchnią musi posiadać cechy charakterystyczne dla danego typu pojazdu (np. pojemność silnika, typ pojazdu, typ silnika itp.). Dla każdego typu pojazdu mogą powtarzać się cechy charakterystyczne, jednak co najmniej jedna powinna być różna na tle innych pojazdów. Niedopuszczalne jest spłaszczenie hierarchii dziedziczenia tak, aby w głównej klasie znajdowały się pola dot. wszystkich typów pojazdów.

Stan serwisu, listy oczekujących oraz magazynów wraz ze wszelkimi danymi dot. osoby, pomieszczeń, przedmiotów, pojazdów, itp. musi być zapisywany po wybraniu z menu odpowiedniej funkcjonalności. Informacje muszę być zapisane w sposób przejrzysty i czytelny dla człowieka z zachowaniem poniższych reguł:

- Stan magazynu powinien być zapisany do pliku ***warehouses.txt***, zaś pomieszczenia wewnątrz piku powinny być posortowane rosnąco według rozmiaru pomieszczenia.
- Zawartość każdego z pomieszczeń powinno być posortowane malejąco według rozmiaru przedmiotu, a jeśli jest taki sam to według nazwy.
- Informacje dot. serwisu samochodowego powinny być zapisane do pliku ***services.txt*** z wyszczególnieniem aktualnych pojazdów w serwisie oraz historii napraw dla każdego z serwisów (wraz z kosztem sumarycznym naprawy każdego z samochodów - wyliczenie na podstawie liczby dni i kosztu jednego dnia serwisu dla danego sposobu - samodzielna/przez mechanika).
- Nie wolno korzystać z żadnego rodzaju serializacji

Zasada działania programu:

- W metodzie ***main*** należy utworzyć serwis samochodowy wraz z co najmniej dziesięcioma gotowymi pomieszczeniami magazynowymi różnego typu i rozmiaru oraz kilka (minimum 5) osób. Ze wstępnie przydzielonymi najmami oraz umiejscowionymi samochodami na miejscach parkingowych i miejscach serwisowych.
- Po uruchomieniu programu, użytkownik powinien mieć możliwość za pośrednictwem konsoli poleceń wywołania wszystkich wspomnianych funkcjonalności. Są to **m.in**:
  - zakończenia programu w dowolnym momencie
  - wybrania którą jest osobą (nie jest potrzebne żadne logowanie, wystarczy wskazanie np. unikalnego numeru osoby)
  - wypisania swoich danych łącznie z wynajętymi pomieszczeniami
  - wyświetlenia wolnych pomieszczeń
  - wynajęcia nowego pomieszczenia, po uprzednim jego wybraniu

- wybrania pomieszczenia które wynajmuje dana osoba oraz wyświetlenia zawartości pomieszczenia
- zaparkowania/włożenia nowych pojazdów/przedmiotów pamiętając, aby nie przepełnić miejsca parkingowego/pomieszczenia
- wyjęcia przedmiotów lub pojazdów.
- rozpoczęcia zgłoszenia serwisowego wybranego pojazdu w wybranym trybie (naprawa przez mechanika lub samodzielna)
- rozpoczęcie naprawy
- wykonania polecenia zapisującego aktualny stan aplikacji do plików

Projekt opiera się o materiał z zakresu PPJ oraz GUI.

Uwaga:

- ***W przypadku otrzymania projektu ze znacznymi brakami w implementacji lub niekompilującego się, wynikiem za taki projekt będzie 0 pkt.***
- ***Brak znajomości dowolnej linii kodu lub plagiat skutkować będzie wyzerowaniem punktacji za ten projekt i/lub niezaliczeniem przedmiotu baz możliwości ubiegania się o poprawę w danym semestrze.***
- ***W ocenie projektu poza praktyczną i merytoryczną poprawnością będzie brana również pod uwagę optymalność, jakość i czytelność napisanego przez Państwa kodu.***
- ***Ważną częścią projektu jest wykorzystanie między innymi: dziedziczenia, kolekcji, interfejsów lub klas abstrakcyjnych, lambda-wyrażeń, typów generycznych, dodatkowych funkcjonalności lub struktur oraz innych elementów charakterystycznych***