
GUI

Projekt 1

Napisz program, który będzie służył do zarządzania przez dewelopera siecią osiedli mieszkaniowych. Deweloper może posiadać wiele osiedli, a w ramach osiedli może istnieć wiele bloków, w których istnieje dowolna (skończona) liczba mieszkań do wynajmu. Każde pomieszczenie może być zamieszkane przez dowolną liczbę osób, o których informacje należy przechowywać (obiekty klasy *Osoba*), jednak zawsze pierwsza osoba, która rozpoczyna najem mieszkania jest osobą odpowiedzialną za opłaty dot. wynajmu.

W ramach najmu mieszkania (tylko jeżeli osoba najmuje mieszkanie w ramach osiedla) jest możliwość dodatkowego najmu zamkniętego miejsca parkingowego, w ramach którego można przechowywać pojazdy, ale również różne przedmioty.

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

Najemca może dowolnie zameldowywać i wymeldowywać osoby z mieszkania, jak i również, jeśli posiada miejsce parkingowe, to może wkładać i wyjmować przedmioty oraz pojazdy. Każda osoba może mieć wynajętych wiele mieszkań i miejsc parkingowych, o ile sumaryczna liczba wynajmowanych mieszkań i miejsc parkingowych przez daną osobę (najemcę) nie jest większa niż 5. Każde pomieszczenie może mieć tylko jednego najemcę w jednym czasie.

Należy zapewnić, aby identyfikatory pomieszczeń były unikalne oraz tworzone automatycznie podczas tworzenia obiektu typu *Mieszkanie* i *MiejsceParkingowe*. Rozmiar powierzchni użytkowej w przypadku oby wspomnianych typów pomieszczeń należy dostarczać podczas tworzenia obiektu pomieszczenia.

Możliwe są dwa sposoby wskazania rozmiaru 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 *File*), które zostaje zapisane do obiektu klasy *Osoba* definiującego konkretnego najemcę.

W zadaniu również należy 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 zostanie odnowiony lub najem zostanie anulowany 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 mieszkania przeprowadzamy eksmisję osób zamieszkałych w tym mieszkaniu.

W przypadku miejsca parkingowego, w pierwszej kolejności sprawdzamy czy znajduje się na miejscu parkingowym pojazd. Jeśli tak, to w związku z przyspieszoną decyzją komorniczą zostaje on sprzedany (z punktu widzenia programu usuwamy obiekt z pomieszczenia), a koszt sprzedaży pokrywa najem pomieszczenia na najbliższe 2 miesiące. Jeśli nie było pojazdu, to utylizacji podlegają wszystkie przechowywane rzeczy na miejscu parkingowym.

Osoba posiada ponad wymienione wcześniej informacje takie dane jak imię, nazwisko, pesel, adres, datę urodzenia.

Jeśli najem będzie chciała rozpocząć osoba z więcej niż trzema zadłużeniami (co najmniej 3 obiekty typu *File*) na przestrzeni najmów na tle całego osiedla, rzucony powinien zostać wyjątek *ProblematicTenantException* z komunikatem – „*Osoba X posiadała już najem pomieszczeń: lista_pomieszczen*”, gdzie X to imię i nazwisko danej osoby, zaś *lista_pomieszczen* definiuje wynajmowane pomieszczenia, które zostały wynajęte.

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

Pojazdy są podzielone na rozróżniające je typy:

- samochód terenowy,
- samochód miejski,
- łódź,
- 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.

W przypadku wkładania każdego przedmiotu lub pojazdu 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*”.

Stan osób zamieszkujących osiedle wraz ze wszelkimi danymi dot. osoby, wynajmowanych pomieszczeń, przedmiotów, itp. musi być zapisywane po wybraniu z menu odpowiedniej funkcjonalności. Informacje zapisane w pliku powinny być zapisane przejrzystie i czytelnie dla człowieka z zachowaniem poniższych reguł:

- Pomieszczenia powinny być posortowane rosnąco według rozmiaru pomieszczenia.
- Zawartość pomieszczenia powinna być posortowana malejąco według rozmiaru przedmiotu jeśli jest taki sam to według nazwy.

Zasada działania programu:

- W metodzie *main* należy utworzyć osiedle wraz z co najmniej dziesięcioma gotowymi pomieszczeniami różnego typu i rozmiaru oraz kilka (minimum 5) gotowych osób. Ze wstępnie przydzielonymi najmami oraz umiejscowionymi przedmiotami na miejscach parkingowych.
- 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
 - włożenia nowych pojazdów/przedmiotów pamiętając, aby nie przepełnić pomieszczenia
 - wyjęcia przedmiotów lub pojazdów.
 - wykonania polecenia zapisującego aktualny stan osiedla do pliku

Projekt opiera się o materiał z zakresu PPJ oraz GUI.

Uwaga:

- *Brak znajomości dowolnej linii kodu lub plagiat skutkować będzie wyzerowaniem punktacji za ten projekt.*
- *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, strumieni, dodatkowych funkcjonalności lub struktur oraz innych elementów charakterystycznych***