Документация за приложение **Home Manager**

# Цел

Основната предназначение е предоставяне на възможност фирмата потребител, да осъществи поддръжка на жилищни сгради. За тази цел са необходими функционалности за определяне на домоуправител (служител на компанията), определляне на таксата поотделно за всяка сграда, поддържане на списък с обслужваните сгради, списък с платежните статуси на всяка една и др.

Тези функционалности са осъществени чрез следните класове: Company, Person, Owner, Employee, Fee, Apartment, Building, HomeManager. Всеки един съдържа два конструктора (дифолтен и с параметри), както и предефиниран метод toString(с изключение на клас HomeManager).

Клас Company

Клас, съдържащ три полета (име на собственика, име на компанията и списък със служителите). Освен аксесорите и мутаторите, са налични още три метода.

**AddEmployee** и **AddEmployees** добавят служител. В AddEmployee се добавя обект на клас AddEmployee, а в AddEmployee се добавя предварително запълнен списък със служителите.

Третият метод отпечатва списъка със служители.

Клас Person

Абстрактен клас, явяващ се базов за други два - Owner и Employee. Съдържа основни описателни характеристики за един човек (име, години, националност, пол). Всички предоставени методи служат за аксесори и мутатори.

Клас Owner

Производен на клас Person, включващ допълнително поле (професия). Всички предоставени методи служат за аксесори и мутатори.

Клас Employee

Производен на клас Person, включва три допълнителни полета (идентификационен номер, максимално допустим брой обслужвани сгради и списък с управляваните сгради). Освен наличните аксесори и мутатори, съществуват методи за добавяне на сграда към списъка и за отпечатване на прилежащите към него сгради.

Клас Fee

Клас, съдържащ две полета - задължителната такса за апартамент и таксата за всеки обитател в него. Всички предоставени методи служат за аксесори и мутатори.

Клас Apartment

Клас, съдържащ основните описателни характеристики за един апартамент (собственик, номер, етаж, брой живущи, съгласие за плащане на таксата по поддръжка). Освен тях, има и още две - зададената от фирмата такса и обща сума от таксите на обитателите.

Освен аксесори и мутатори, още два метода допълват класа. Първият SetPaymentFeе задава изискваната от фирмата такса (обект на клас Fee). Методът се извиква в метода SetRequiredFee на клас Building.

Вторият PayFee проверява дали собственикът е съгласен да плати изискваната от фирмата такса. При положителен отговор се изчислява общата за апартамента сума (задължителна за апартамент + такса за обитател \* броя на обитателите в дадения апартамента).

Клас Building

Клас, съдържащ основните описателни характеристики за един апартамент (адрес, година на строителство, застроена площ, обитаема площ, общи части, брой етажи, брой апартаменти, брой обитатели, строителен материал). Към тях са добавени едно поле (за мениджър от клас Employee) и две колекции - списък с апартаментите в сградата и платежният статус (относно таксата) на всеки апартамент. Освен аксесорите и мутаторите, има още няколко метода.

SetBuildingManager - задаване на домоуправителя (клас Employee)

SetRequiredFee – апартамент от списъка извиква аксесора си SetPaymentFee и подава като параметър изискваната от фирмата такса.

GetBuildingResidentsAmount - броят на обитателите се определя като сума от обитателите във всеки апартамент

AddApartment и AddApartments се добавя апартамент. В AddApartment се добавя обект от AddApartment, а в AddApartments предварително запълнен списък с апартаменти.

ShowOwnersList отпечатва списъка с апартаменти.

Клас HomeManager

Клас, съдържащ основните описателни характеристики (фирма, задължителна такса за апартамент, такса за обитател). Допълнителните полета са (изчислената такса за апартамент, сбора на таксите на всички апартаменти, сума на събраните такси, брой на обитатели във всяка сграда) и четири колекции (списък със управлявани сгради, списък с домоуправителите, списък на апартаментите в дадената сграда, списък на платежния статут за всеки апартамент).

Освен аксесори и мутатори, налични са и следните методи.

Методът SetHomeManager е overload-нат. В едната си версия избира домоуправител, чието име е зададено от фирмата потребител, а в другата - се избира, в зависимост дали той е достигнал броят на максимално обслужвани сгради. Ако не е, името на служителя се отпечатва. В противен случай изхвърля изключение от тип MaxBuildingsAmountException.

SetFloorFeeIncrement и SetResidentFeeIncrement - задават допълнително повишение на таксите, в зависимост от броя на етажите и обитателите

SetCalculatedApartmentFee - задава на сградата изискваната такса

SetPaymentStatus и SetStatus - отбелязват променения платежен статут на дадения апартамент, в списъка с апартамент

GetCollectedFees - сумира всички платени такси

AddHomeManagers - добавя предварително създаден списък с служители

AddHomeManagerToList - добавя служител (клас Employee)

AddBuildingToList - добавя сграда към списъка с обслужвани сгради

ShowRequiredFee - отпечатва общата сума, която сградата трябва да предостави

ShowManagedBuildings - отпечатва списъка със сградите

ShowPaymentStatus - отпечатва всеки апартамент, неговата такса и платежният му статут  
 ShowResidentsAmount - отпечатва броя на обитателите в дадената сграда

Начин на работа

Фирмата потребител въвежда необходимите данни за таксата, максималния брой управлявани от служител сгради и (евентуално) конкретен служител за домоуправител.

**Изготвил: Зорница Йоханова Йоханова**

**F85824**