

CSCB525 Приложно програмиране с Java

Домашна работа № 8 / седмица 11

Unit Testing. Concurrent API

В компания се сключват сделки за продажба на недвижими имоти. За тази цел, е реализиран клас сделка (**Deal**), в който има пореден номер (уникален) на сделката и **Map<Имот, Цена>**. Имотите (**Estate**) се определят от пореден номер (уникален) на имота, площ и категория (**Category**).

Задача 1. Да се напишат unit тестове, с които се проверява дали методите в клас **Deal** работят правилно. Необходимо е да се реализират тест-методи, които правят следните проверки:

- a) Да се провери дали метод `dealTotalValue()` връща правилната стойност при списък с поне един имот.
- b) Да се провери дали метод `dealTotalValue()` връща правилната стойност при празен списък с имоти.
- c) Да се провери дали метод `highestPrice()` връща правилната стойност при списък с поне един имот.
- d) Да се провери дали метод `highestPrice()` връща правилната стойност при празен списък с имоти.
- e) Да се провери дали метод `getEstatePrice()` връща правилната стойност в случай, че имотът, който се предава като аргумент е в списъка с имотите по сделката.
- f) Да се провери дали метод `getEstatePrice()` хвърля изключение `NoSuchEstateException`, в случай че имотът, който се предава като аргумент **не** е в списъка с имотите по сделката.
- a) Да се провери дали метод `isBigDeal()` връща `true`, ако общата стойност по сделката е по-голяма от число, което се предава като аргумент на метода.
- a) Да се провери дали метод `increasedPriceList()` връща `Map<Имот, Цена>`, в който цените на имотите са по-високи от тези в сделката с процент, който се предава като аргумент на метода. Пример: ако списъкът с имоти в сделката е: Имот1 -> 2000; Имот2 -> 1000; Имот3 -> 3000, при изпълнение на метод `increasedPriceList(10)` се връща следния списък: Имот1 -> 2200; Имот2 -> 1100; Имот3 -> 3300.

Задача 2. Клас `MakingDeal` се използва за осъществяване на всяка сделка в отделни нишки. Той трябва да имплементира интерфейс `Callable`. Метод `call()` трябва да връща общата стойност по сделката. В `main()` метода на клас `Main` да се използва `ExecutorService` за създаване на `pool` (съвкупност) от 6 сделки чрез клас `MakingDeal`. Да се използва семафор (`semaphore`), за да се осигури, че едновременно ще бъдат пуснати само 2 нишки.

Класовете `Estate` (Имот), `Category` (Категория на имота), `Deal` (Сделка), `MakingDeal` (Осъществяване на сделка) и `NoSuchEstateException` (Изключението за несъществуващ имот) са прикачени към заданието.