

Задачи за домашно към упражнение 6

Задача 1

Подберете подходящи модификатори за достъп в задачата и поставете сорс файловете в пакет `bg.tu_varna.sit.task1`

Да се създаде програма за книжарница.

За целта са необходими:

- Клас `Author` който се описва с име (`firstName`), фамилия (`lastName`) и държава (`country`). Авторите трябва да могат да се сравняват по две имена и държава, и да се представят текстово във формат `<lastName>, <firstName>`

- Интерфейс `Margin` с метод `calculateMargin()`, който изчислява надценката на дадена книга;

- Абстрактен клас `Book`, който се описва с заглавие (`title`), автор (`author`), година на издаване (`publishingYear`), цена (`price`) и вид на кориците (твърди (`true`)/меки (`false`)) (`hasHardCover`), като имплементира интерфейс `Margin`. Поведението на книгата включва `getFinalPrice()`, който изчислява и връща крайната цена на книгата (сумата на цената и надценката). Тато текст книгата се представя като `< author>: <title>`

- Клас `CrimeNovel`, наследяващ `Book`. Надценката се изчислява като 2% от цената ако книгата е с твърди корици и като 1% от цената в случай на меки корици;

- Клас `PoetryBook`, наследяващ `Book`. Надценката се изчислява като 1% от цената ако книгата е с твърди корици и е издадена след 2000 година, и като 5% в останалите случаи;

- Клас `SciFiNovel`, наследяващ `Book`. Надценката се изчислява като 9% от цената ако книгата е с меки корици и 12% от цената в останалите случаи.

Създайте клас `Bookstore`, който има поле масив от книги (`books`) и поведение:

- `calculateTotalBookPrice()`, който изчислява и връща цената на всички книги;

- `calculateAverageBookPrice()`, който изчислява и връща средната цена на книгите в книжарницата.

Задача 2

Подберете подходящи модификатори за достъп в задачата и поставете сорс файловете в пакет `bg.tu_varna.sit.task2`

Да се създаде програма, описваща агенция за недвижими имоти.

За целта са необходими:

- Интерфейс `Commission` с поведение за изчисляване и връщане на комисионната за даден имот, `calculateCommission`

- Абстрактен клас `Property`, който се описва с площ(`area`), цена(`price`) и дали имота се отдава под наем или се продава (`isForRent`), като имплементира интерфейс `Commission`.

- Клас `Apartment`, наследник на `Property`, който се описва с брой стаи (`numberOfRooms`) и номер на етаж (`floor`).

Комисионната се изчислява като 15% от цената ако апартаментът е за отдаване под наем; 10% ако площта му е по-малка от 60 кв.м. и 7% от цената в останалите случаи;

- Клас `House`, наследник на `Property`, който се описва с брой етажи (`numberOfFloors`) и наличието на градина или не (`hasGarden`).

Комисионната се изчислява като 8% ако къщата се отдава под наем и има градина; 5% ако къщата е за продажба и площта е по-малка от 100 кв.м. и 3% от цената в останалите случаи;

- Клас `Office`, наследник на `Property`, който се описва с наличието на паркинг (`hasParkingLot`) и брой стаи (`numberOfRooms`).

Комисионната се изчислява като 18% от цената ако офиса е за отдаване под наем, 15% ако има паркинг и повече от две стаи, и 11% във всички останали случаи;

- Клас `Shop`, наследник на `Property`. Комисионната се изчислява като 2% ако магазинът е за отдаване под наем и площта му е по-малка от 50 кв.м., 1% ако площта му е над 100 кв.м. и 6% във всички останали случаи.

Създайте клас `RealEstateAgency`, който има полета за име (`name`) и масив от недвижими имоти (`properties`). Поведението на класа е следното:

- `calculateTotalExpectedCommission()`, изчислява и връща очакваната комисионна от всички предлагани имоти;

- `getPropertiesForSale()`, намира и връща броя недвижими имоти, предлагани от агенцията за продажба.

Задача 3

Подберете подходящи модификатори за достъп в задачата и поставете сорс файловете в пакет `bg.tu_varna.sit.task3`

Да се създаде програма за магазин. За целта са необходими:

- Интерфейс `Delivery`, който дефинира поведение дали дадената стока се нуждае от доставка `needsDelivery()`, който указва;

- Интерфейс `Food`, указващ дали дадената стока е хранителна, разширява интерфейса `Delivery`;

- Интерфейс `Drink`, указващ дали дадената стока е напитка, разширява интерфейса `Delivery`;

- Абстрактен клас `Item`, описващ се с име (`name`), срок на годност в дни (`daysToExpire`) и налично количество (`availableQuantity`). Класа имплементира `Delivery`.

- Клас `Bread`, наследяващ `Item` и го допълва с съдържание на бяло брашно (`whiteFlourPercentage`). Имплементира `Food`.

Има необходимост за доставка ако срокът за годност е по-малък от 5 дни и хлябът е със съдържание на бяло брашно над 75 или са останали по-малко от 10 бройки;

- Клас `Cheese`, наследяващ `Item` и го разширява с грамаж на опаковката (`gramsPerPackage`), имплементира `Food`.

Има необходимост от доставка ако са останали по-малко от 10 бройки сирене в опаковки до 400 грама;

- Клас `Milk`, наследяващ `Item` и го разширява с указващ процента масленост (`fatPercentage`).

Има необходимост от доставка ако млякото е трайност до 15 дни, останали са по-малко от 10 броя и маслеността е по-голяма от 2%;

- Клас `Water`, наследяващ `Item` и го разширява с обща минерализация (`mineralContent`),

Има необходимост от доставка ако са останали по-малко от 30 броя или ако срокът на годност изтича след по-малко от 20 дни.

Създайте клас `Shop`, който се описва с масив от стоки (`items`), Поведението на класа е:

- `getItemCountDelivery()`, изчислява и връща броя стоки, които се нуждаят от доставка;

- `getAverageDaysToExpire()`, изчислява и връща средния срок на годност за стоките в магазина;

- `getAvailableDrinksCount()`, който намира и връща колко са наличните напитки в магазина.

Задача 4

Подберете подходящи модификатори за достъп в задачата и поставете сорс файловете в пакет `bg.tu_varna.sit.task4`

Съставете програма, която има за цел да изчислява добива на риба от плавателните водни басейни на дадена територия.

Създайте клас риба (Fish) с полета наименование (name) и количество от нея по експертна оценка, тона (quantity). Класът се разширява от:

- ядлива риба (EdibleFish), която допълва информацията с процент от популацията, която е годна за улов (възрастни индивиди) (percentOfYield);
- негодна за консумация риба (NonEdibleFish), където допълнително като текст се описва причината тя да е неизползваема или да е заплаха за здравето (threat).

Създайте интерфейс Usage, който да съдържа boolean метод isProductabe().

Интерфейсът да се имплементира от клас воден басейн (WaterBody). Водният басейн се описва с наименование (name), дълбочина (depth) и масив от риби, които го обитават.

Имплементира метода isProductabe(), където за потенциално продуктивен се счита басейн, който е обитаван от поне една годна за консумация риба с количества, които разрешават нейния улов (над 10 тона).

Да се добави абстрактно поведение isFloaty(), чието предназначение е да определи дали водният басейн е плаваем или не.

Да се допълни с метод calculateProduction(), който да изчислява възможно най-големите количества риба, които законно могат да се уловят от плавателния басейн.

WaterBody се наследява от:

- Езеро (Lake) с широчина (width) и дължина (length). Плаваемо е, ако дълбочината му е поне 5 м., а широчината и дължината – поне 1000 м.
- Река (River) със скорост на течението (km/h) (speed). Плаваема е, ако дълбочината е поне 3 м. и скоростта на течението не надвишава 30 km/h.

Задача 5

Подберете подходящи модификатори за достъп в задачата и поставете сорс файловете в пакет `bg.tu_varna.sit.task5`

Съставете програма, с която да управлявате наличностите от напитки в бар.

Създайте интерфейс `Delivery`, който да съдържа два метода:

- `needOfDelivery()`, с който да се установява има ли (`true`) или няма (`false`) нужда от доставка от дадена напитка;

- `deliver()`, с помощта на който да се доставят нови количества от съответната напитка. Приема като параметър доставените количества.

Добавете втори интерфейс `Serving`, който съдържа метод `serve()` за сервиране на единица от съответната напитка. Приема като параметър броя единици, поръчани от съответната напитка. Например при поръчка на 5 чаши натурален сок, броят на единиците е 5.

Създайте клас `Drink`, който имплементира двата интерфейса. Всяка напитка има наименование (`name`); налично количество, литра (`quantity`) и обем на една единица за сервиране, литра (`serveQuantity`). Включва имплементацията на методите:

- `deliver()` – увеличава наличните количества с подадените като параметър доставени количества;

- `serve()`- при подадени като параметър единици за сервиране извършва проверка дали има достатъчна наличност от напитката. Ако да, те се сервират директно на клиентите. Ако не, се извършва проверка колко единици са налични и само те се сервират. (сервирането става като се извадят сервираните количества от общото количество)

В текстовото представяне на напитките включете тяхното наименование, налични количества и необходимост от доставка на нови количества.

`Drink` се разширява от класове безалкохолна напитка (`SoftDrink`), която допълва информацията за напитката с това дали е без захар (`isSugarFree`) и алкохолна напитка (`AlcoholicBeverage`), която допълнително описва и обемния процент на съдържащия се в нея алкохол (`vol`). Минималното количество от дадена безалкохолна напитка, които барът се стреми да поддържа, е 10 литра, а от алкохолна – 5 литра. Под тези наличности е необходима доставка.