Задачи за домашно към упражнение 9

Задача 1

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

Да се състави програма за резервиране на хотелски стаи. За целта са необходими:

- Интерфейс ценови калкулатор (PriceCalculator) с методи за изчисляване и връщане стойностите на престоя (calculateStayPrice), възможната отстъпка(discount) и крайната цена на резервация (calculateReservationPrice);

- Енумерация за изложение (Exposure) със стойности морска гледка (SEA_VIEW) и изглед към парк (PARK_VIEW);

- Абстрактен клас стая (Room), имплементиращ интерфейс ценови калкулатор. Класът има атрибути цена на нощувка (pricePerDay), изложение (exposure) и дали стаята е заета (available). Дефинирайте конструктор по цена на нощувка и изложение, като по подразбиране стаята е свободна. Дефинирайте методи за достъп и метод за модификация на атрибута, указващ дали стаята е свободна или не. Имплементирайте метода, изчисляващ крайната цена на резервация като разлика от стойността на престоя и възможната отстъпка;

- Клас единична стая (SingleRoom), който разширява стая с атрибут брой дни (days). Дефинирайте конструктор по всички полета, методи за достъп и текстово описание. Цената на резервацията се изчислява като произведение от броя дни и цената за нощувка; при резервация за повече от 3 дни се прави 10% отстъпка;

- Клас двойна стая (DoubleRoom), който разширява стая с атрибути брой дни(days) и дали ще има деца (hasChildren). Дефинирайте конструктор по всички полета, методи за достъп и метод за текстово описание. Цената на резервацията се изчислява като произведение от броя дни и цената на нощувка, увеличена с 10. Има отстъпка 15% от цената ако ще има деца и нощувките ще са повече от 5;

- Клас хотел (Hotel), който има като атрибут списък от стаи и конструктор по подразбиране. Методи:

-- за добавяне на стая (addRoom);

-- за създаване на резервация (createReservation), който проверява дали има свободна стая от желания тип и маркира първата срещната такава като заета;

-- изчисляване и връщане стойността на отстъпката от всички заети стаи (calculateBookedRoomsDiscount);

-- изчисляване и връщане средната стойност от цената на свободните стаи с морски изглед (calculateAveragePriceOfAvailableRoomsWithSeaView);

-- за текстово описание.

Дефинирайте клас Application с главна функция и тествайте описаните функционалности.

Задача 2

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

Да се състави програма за книжен каталог. За целта са необходими:

- Изключение за невалидни данни (InvalidDataException);

- Клас автор (Author), който имплементира интерфейс Comparable. Класът е с атрибути за име (firstName), фамилия (lastName) и държава (country). Дефинирайте конструктор по всички полета, методи за достъп, текстово описание и равенство (по всички полета). Интерфейсният метод сравнява по име, фамилия и държава;

- Клас книга (Book), имплементиращ интерфейс Comparable, който има атрибути за заглавие (title), автор (author), година на издаване (publishingYear) и налично количество (availableQuantity). Дефинирайте методи за достъп, метод за модификация на наличното количество, метод за равенство (по автор и заглавие), метод за сравнение (по автор и заглавие) и метод за текстово описание;

- Клас книжен каталог (BookCatalogue), който има като атрибут колекция от уникални книги и конструктор по подразбиране.

Методи:

-- за добавяне на книга (addBook);

-- за премахване на книга (removeBook);

-- за заемане на книга (borrowBook), който проверява дали има такава книга в каталога и променя наличното количество.

Ако заеманата книга е последна бройка, тя се премахва от колекцията;

-- за връщане на книга (returnBook), който увеличава наличното количество. Ако книгата не присъства в каталога, тя се добавя като първа бройка;

-- за изчисляване и връщане на брой книги по зададен автор (countBooksByAuthor);

-- за сортиране на книгите по автор (sortCatalogueByAuthor);

-- за сортиране на книгите по налично количество (sortCatalogueByAvailableQuantity);

-- за сортиране на книгите по заглавие (sortCatalogueByTitle);

-- за намиране и връщане на броя книги, издадени след дадена година (countBooksPublishedAfterYear);

-- за текстово описание.

Дефинирайте клас Application с главна функция и тествайте описаните функционалности.

Задача 3

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

Да се състави програма за книжен каталог. За целта са необходими:

- Изключение за невалидни данни (InvalidDataException);

- Енумерация специалност (Speciality) със стойности СИТ (SIT), КСТ (CST), КТ (CCT) и А (A).

- Клас човек (Person) с атрибути за име (firstName), фамилия (lastName) и възраст (age). Дефинирайте конструктор по всички полета, методи за достъп, текстово описание и равенство (по всички полета);

- Клас студент (Book), имплементиращ интерфейс Comparable, който има разширява клас човек с атрибути факултетен номер (fNumber), специалност (speciality), курс (course) и успех (grades). Дефинирайте методи за достъп, методи за модификация с валидиране на курса и успеха, метод за равенство по факултетен номер и метод за текстово описание. Интерфейсният метод е също по факултетен номер;

- Клас факултет (Faculty), който има като атрибути име на факултета (facultyName) и колекция от студенти (students).

Дефинирайте конструктор по име и метод за достъп до името. Методи:

-- за добавяне на студент (addStudent);

-- за изчисляване и връщане средния успех на студентите във факултета (calculateAverageGrades);

-- за намиране и връщане броя студенти в дадена специалност (getNumberOfStudentsBySpeciality);

-- за намиране и връщане на студентите с отличен успех от факултета (getStudentsWithExcellentGrades);

-- за намиране и връщане броя студенти в зададен курс (getNumberOfStudentsByCourse);

-- за текстово описание.

Дефинирайте клас Application с главна функция и тествайте описаните функционалности.

Задача 4

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

Да се състави програма за недвижими имоти. За целта са необходими:

- Интерфейс ценови калкулатор (PriceCalculator) с метод, който изчислява цената на имота (calculate);

- Енумерация за тип на имота (PropertyType) със стойности ново строителство (NEW) и старо строителство (OLD);

- Енумерация за изложение на имота (Exposition) с възможните осем стойности;

- Абстрактен клас имот (Property) с атрибути за площ (area), базова цена (basePrice) и тип (propertyType). Класът имплементира интерфейс ценови калкулатор и интерфейс Comparable. Дефинирайте конструктор по всички полета, методи за достъп, метод за равенство (по всички полета) и метод за текстово описание. Интерфесният метод е по площ;

- Клас апартамент (Apartment), който разширява имота с атрибути за етаж (floorNumber), брой стаи (numberOfRooms), изложение (exposition) и наличие на паркомясто (hasParkingPlace). Дефинирайте конструктор по всички полета, методи за достъп, метод за равенство по всички полета и метод за текстово описание. Цената на даден апартамент се изчислява като:

-- базовата цена, увеличена с 25%, ако апартамента е ново строителство, има изложение юг/запад/югозапад, площта му е повече от 50 кв.м. и има паркомясто;

-- базовата цена, увеличена с 20%, ако апартамента е ново строителство и има изложение юг/запад/югозапад;

-- базовата цена, увеличена с 15%, ако апартамента е ново строителство;

-- базовата цена, увеличена с 18%, ако апартамента е старо строителство и има паркомясто;

-- базовата цена, увеличена с 12% във всички останали случаи;

- Клас къща (House), който разширява имота с атрибути брой етажи (numberOfFloors) и наличие на градина (hasGarden).

Дефинирайте конструктор по всички полета, методи за достъп, метод за равенство по всички полета и метод за текстово описание. Цената на дадена къща се изчислява като:

-- базовата цена, увеличена с 20%, ако къщата е ново строителство, има градина и е с повече от един етаж;

-- базовата цена, увеличена с 10%, ако къщата има градина;

-- базовата цена, увеличена със 7% във всички останали случаи;

- Клас агенция за недвижими имоти (RealEstateAgency), който има като атрибут списък от имоти (properties) и конструктор по този атрибут. Методи:

-- за добавяне на имот (addProperty);

-- за изчисляване и връщане на цената на предлаганите имоти (calculatePriceOfAllProperties);

-- за сортиране на имотите по площ (sortPropertiesByArea);

-- за сортиране на имотите по цена (sortPropertiesByPrice);

-- за намиране и връщане броя имоти от даден тип (getNumberOfPropertiesByType);

-- за намиране и връщане броя предлагани къщи (getNumberOfAvailableHouses);

-- за намиране и връщане на най-скъпия предлаган апартамент (getMostExpensiveApartment);

-- за изчисляване и връщане средната цена на предлаганите къщи (calculateAverageHousePrice);

-- за текстово описание.

Дефинирайте клас Application с главна функция и тествайте описаните функционалности.

Задача 5

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

Да се състави програма за банка. За целта са необходими:

- Интерфейс лихвен калкулатор (InterestCalculator) с метод, който изчислява стойността на лихвата по дадена сметка (calculateInterestValue);

- Енумерация за валута (Currency) със стойности лев, долар, евро и паунд;

- Енумерация за тип на сметката (AccountType) със стойности лична (PERSONAL) и корпоративна (CORPORATE);

- Изключение за невалидни данни (InvalidDataException);

- Абстрактен клас сметка (Account) с атрибути за номер (id), тип (type), валута (currency) и наличност (balance), който имплементира интерфейс лихвен калкулатор. Дефинирайте конструктор по всички полета, методи за достъп и метод за равенство (по номер);

- Клас депозитна сметка (DepositAccount), който разширява сметката. Дефинирайте конструктор по всички полета и метод за текстово описание. Лихвата се изчислява като:

-- 9% от наличността при лична сметка;

-- 7% от наличността при сметка в лева;

-- 5% от наличността при сметка в долари;

-- 2% от наличността при сметка в евро;

-- 1% от наличността в останалите случаи;

- Клас сметка за заплата (SalaryAccount), който разширява сметката. Дефинирайте конструктор по всички полета и метод за текстово описание. Лихвата се изчислява като:

-- 0 за корпоративна сметка;

-- 5% от наличността за сметка в лева;

-- 3% от наличността за сметка в долари;

-- 2% от наличността за сметка в евро;

-- 1% от наличността в останалите случаи;

- Клас спестовна сметка (SavingsAccount), който разширява сметката. Дефинирайте конструктор по всички полета и метод за текстово описание. Лихвата се изчислява като:

-- 15% от наличността за лична сметка в лева;

-- 5% от наличността за сметка в лева;

-- 10% от наличността за лична сметка в долари;

-- 1% от наличността в останалите случаи;

- Клас банка (RealEstateAgency), който има като атрибут колекция от сметки. Методи:

-- за създаване на сметка (createAccount), който добавя сметката към колекцията ако тя не съществува;

-- за затваряне на сметка (closeAccount);

-- за намиране и връщане на броя сметки за заплата, открити в банката (calculateNumberOfSalaryAccounts);

-- за намиране и връщане на средната лихва за сметки в дадена валута (calculateAverageInterestByCurrency);

-- за намиране и връщане на най-високата лихва (findHighestInterest) с отчитане на валутния курс;

-- за текстово описание.

Дефинирайте клас Application с главна функция и тествайте описаните функционалности.