Задачи за домашно към упражнение 8

Задача 1

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

Създайте приложение за нуждите на натуропатичен медицински кабинет, в който се предлага натурална добавка за укрепване на съпротивителните сили на организма. В него се обслужват възрастни и деца над 3 години, но има идея да се развие и в насока на някои домашни любимци.

Създайте клас, описващ дете (Child) с данни за име (name), възраст (age) и телесно тегло (weight). При създаването на обект от класа, ако посочената възраст е под 3 г., да се хвърля изключение InvalidAgeException. Текстовото представяне на обекта да бъде във формат:

> Име: <име на детето>,
> Възраст: <възраст на детето>

Създайте клас, описващ възрастен (Adult) с данни за име (name) и номер на здравната осигуровка (assuranceNumber). Текстовото представяне на обекта да бъде във формат:

> Име: <име>,
> Номер на здравна осигуровка: <номер>

Предвидете клас за медицински преглед (Medical), който да бъде описан от параметризиран пациент (patient) и общо състояние на пациента (condition), като изборът (Condition) е между възможностите good, stable и damaged. Като поведение предвидете методи за:

- Изчисляване на подходяща дозировка в милилитри на хранителната добавка (calculatePotion). Ако пациентът е дете, се предвиждат 0.250 мл. на всеки килограм телесно тегло, като резултатът се закръгля до цяло число. При останалите пациенти дневната дозировка е 25 мл. Ако състоянието на пациента е увредено, тя се удвоява. Методът връща предписаното количество.
- Издаване на рецепта (getPrescription). Извежда като текст данните на пациента от текстовото му представяне и предписаното количество от хранителната добавка във формат:

  > Пациент:
  > <данни на пациента>
  > Дозировка: <посочва се> мл.

- Издаване на купон за следващ преглед (getCoupon), който дава възможност за 25% отстъпка. Методът връща като текст „Издаден купон на: <посочва се>”, където може да бъде зададено име, номер на съществуващ пациент от регистъра на кабинета (цяло петцифрено число) или да бъде оставен неименуван, като за целта като параметър се подаде символа „-“.

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.

Създайте програма, която ще подпомогне управлението на архива на документацията във фирма X. Документацията се архивира на хартиен носител, диск, USB памет и други носители. Носителите се съхраняват физически в помещение с подходящи условия за опазване на тяхната цялост и пригодност. Помещението е разделено на секции, като във всяка от тях се съхраняват документи на еднотипни носители.

Създайте клас документ, съхранен на хартиен носител (PaperDocument) с описание за заглавие на документа (title), дата на съставяне (съхранете като текст във формат yy-mm-dd) (dateCreated), брой страници (pages) и достъп (access), като вариантите на достъп (Access) са free и restricted. Документите на хартиен носител да се сравняват по тяхното заглавие и дата на съставяне.

Съставете клас за съхранение на документ на диск (DiscStorage), като всеки диск има идентификатор (id) (цяло число) и текстово описание на съдържанието му (content). Дисковете се сравняват по техния идентификатор.

Създайте параметризиран клас секция (Section) с полета наименование (title) и параметризиран масив от документи (documents). Добавете метод за търсене на документ в секцията (findDocument), който приема параметризиран обект и връща текст „Открит“ или „Не е намерен“, в зависимост от резултата.

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 талон за отстъпка.

За целта са необходими:

- Интерфейс отстъпка (Discount) с методи за изчисляване на общата отстъпка (calculateTotalDiscount) и за изчисляване на средната отстъпка (calculateAverageDiscount);

- Енумерация за възрастова група (AgeGroup) със стойности дете (CHILD), младеж (TEENAGE), възрастен (ADULT) и пенсионер (PENSIONER);

- Клас Потребител (Person) с атрибути име (name), възрастова група (ageGroup) и цена на закупения продукт (productPrice).

Дефинирайте конструктор по всички полета и методи за достъп. Създайте и метод, който изчислява и връща базова отстъпка (calculateBaseDiscount), който връща 8 процента от цената на продукта, ако потребителят е дете или пенсионер, и 3 процента в останалите случаи;

- Клас отстъпка за храна (FoodDiscount), който имплементира интерфейс отстъпка и има като атрибути масив от потребители (people) и процент на отстъпка за храна (foodDiscountRate). Дефинирайте конструктор по всички полета и методи за достъп.

Имплементирайте интерфейсните методи;

- Клас отстъпка за напитка (DrinkDiscount), който имплементира интерфейс отстъпка и има като атрибути масив от потребители (people) и процент отстъпка за напитка (drinkDiscountRate). Дефинирайте конструктор по всички полета и методи за достъп.

При имплементацията на интерфейсните методи се отчитат само потребителите от възрастови групи възрастен и пенсионер;

- Клас отстъпка за игра (GameDiscount), който имплементира интерфейс отстъпка и има като атрибути масив от потребители (people), процент отстъпка (discountRate) и процент отстъпка за възрастни (discountRateForAdults). Дефинирайте конструктор по всички полета и методи за достъп. Имплементирайте интерфейсните методи.

Създайте генерик клас за талон за отстъпка (Coupon). Дефинирайте в него генерик методи за извеждане на общата отстъпка (displayTotalDiscount) и за извеждане на средната отстъпка (displayAverageDiscount).

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 изчисляване на лихви върху банкови сметки.

За целта са необходими:

- Интерфейс лихвен калкулатор (InterestCalculator) с метод за изчисляване на лихвата (calculateAccountInterest);

- Енумерация за валута (Currency) със стойности лев (BGN), долар (USD) и евро (EUR);

- Клас клиент (AccountHolder) с атрибути име (firstName), фамилия (lastName) и възраст (age). Дефинирайте конструктор по всички полета и методи за достъп;

- Абстрактен клас сметка (Account), имплементиращ интерфейса лихвен калкулатор. Класът има атрибути за клиент (accountHolder), валута (currency) и наличност (balance). Дефинирайте конструктор по всички полета и методи за достъп;

- Клас депозитна сметка (DepositAccount), който наследява клас сметка. Интерфейсният метод връща 5% от наличността, ако сметката е в лева, 2% от наличността, ако сметката е в долари и 1% в останалите случаи;

- Клас спестовна сметка (SavingsAccount), който наследява клас сметка. Интерфейсният метод връща 8% от наличността, ако клиента е с възраст над 62 години и сметката е в лева; 5% от наличността, ако сметката е в лева, 2.5% от наличността, ако сметката е в долари и 0.5% в останалите случаи;

- Клас сметка за заплата (SalaryAccount), който наследява клас сметка. Интерфейсният метод връща 8% от наличността, ако възрастта на клиента е между 25 и 62 гоини, и 4% в останалите случаи.

Създайте генерик клас за лихва (Interest). Дефинирайте в него генерик метод за извеждане на лихвата за дадената сметка (displayAccountInterest).

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 прогнозиране на победител от спортна среща.

За целта са необходими:

- Интерфейс прогнозен калкулатор (PredictionCalculator) с метод за предвиждане и връщане на името на победителя (predictWinner);

- Интерфейс резултати (Results) с метод, който изчислява и връща вероятността за победа (winProbability);

- Енумерация за категория отбор (TeamRank) със стойности силен отбор (TOP_TEAM), отбор от средата (AVERAGE_TEAM) и слаб отбор (BOTTOM_TEAM);

- Енумерация за резултат от последната среща (LastGameResult) със стойности победа (WIN), загуба (LOSS) и равенство (DRAW);

- Абстрактен клас отбор (Team), имплементиращ интерфейс резултати. Класът има атрибути за име (name), категория (teamRank) и резултат от последната среща (lastGameResult). Дефинирайте конструктор по всички полета и методи за достъп;

- Клас домакин (HomeTeam), който наследява отбор. Интерфейсният метод връща:

-- ако отборът е силен: 90% при резултат от последната среща победа, 65% при резултат от последната среща равенство и 55% при загуба в последната среща;

-- ако отборът е в средата на класирането: 70% при победа в последната среща, 48% при равенство в последната среща и 35% при загуба в последната среща;

-- в останалите случаи 60% при победа в последната среща, 40% при равенство в последната среща и 20% при загуба;

- Клас гост (AwayTeam), наследяващ клас отбор. Интерфейсният метод връща:

-- ако отборът е силен: 80% при резултат от последната среща победа, 60% при резултат от последната среща равенство и 45% при загуба в последната среща;

-- ако отборът е в средата на класирането: 60% при победа в последната среща, 40% при равенство в последната среща и 30% при загуба в последната среща;

-- в останалите случаи 55% при победа в последната среща, 38% при равенство в последната среща и 15% при загуба;

- Клас футболна среща (Football), имплементиращ интерфейс прогнозен калкулатор с атрибути домакин (homeTeam) и гост (awayTeam). Дефинирайте конструктор по всички полета. Интерфейсният метод връща домакина при вероятност за негова победа повече от 80%, равенство (draw) при вероятност за победа на домакина над 50% и госта в останалите случаи;

- Клас волейболна среща (Volleyball), имплементиращ интерфейс прогнозен калкулатор. Класът има атрибути за домакин (homeTeam), гост (awayTeam), дали е игран тайбрек в последния мач на домакина (lastGameTieBreakHome) и дали е игран тайбрек в последния мач на госта (lastGameTieBreakAway). Интерфейсният метод връща:

-- ако е игран тайбрек в последния мач на домакина: госта, ако вероятността за победа на домакина е до 60%, в противен случай се връща домакина;

-- ако е игран тайбрек в последния мач на госта: домакина, ако вероятността за победа на домакина е до 60%, в противен случай се връща госта;

-- в останалите случаи се връща госта, ако вероятността за победа на домакина е до 70% и домакина в останалите случаи.

Създайте генерик клас за прогноза (MatchPrediction). Дефинирайте в него генерик метод за извеждане на прогнозния победител (displayPredictedWinner).

Дефинирайте клас Application с главна функция и тествайте описаните функционалности.