

Задачи за домашно към упражнение 5

Задача 1

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

Да се реализира клас Град (City), описващ име (name) и пощенски код (postalCode) (положително цяло число).

Да се реализира клас Лична карта (IdentificationCard), описващ номер (number), Година на издаване (year), ЕГН (egn), Град на издаване (city). Като предоставя поведение за, издадена от (publishedBy), връща: МВР Името на града на издаване и валидна до (validUntil), връща: Година на издаване + 10 години

Да се реализира клас Person, описващ лични данни: Първо (firstName), Второ (secondName) и Трето (thirdName) име. Имплементира метода toString извеждащ имената във формат: Трето, Първо Второ, и методи за четене и запис на полетата.

Да се реализира клас Adult, разширяващ класа Person с Лична карта (IdentificationCard). Имплементира метода toString връщащ информацията във форма:

Трето, Първо Второ

Лична карта номер:

Издадена от:

Подберете подходящи модификатори за достъп при всеки клас.

Задача 2

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

Реализирайте клас страница (Paper), който се описва с процент на запълване (fillRate) и имплементира родителския метод clone.

Да се реализира клас Машина (Machine) която има цена (price) и поведение създаване на печатен лист (createTracePaper), който приема като параметър брой листи за отпечатване и връща като резултат масив с ново създадени страници.

Машината се наследява от клас Принтер (Printer) и се разширява с брой на страници (numberOfPages), които се принтират за 1 минута и съдържание за принтиране (content) метода приема текст, който трябва да се принтира. Класа Printer имплементира метода createTracePaper като разделя текста за принтиране по 2500 символа на страница. Процента на запълване на страница се определя от броя печатаеми символи, непечатаемите символи са: нов ред; таболация; спейс. Параметъра на метода createTracePaper се използва за това колко екземпляра да бъдат направени от един лист.

Създайте клас Копирна машина (Duplicator), който разширява машината с количество хартия (paperCount) и съдържание за принтиране (content) метода приема страница за копиране. Класа Duplicator имплементира метода createTracePaper, като създава нова обект от класа Paper с метода clone(). Ако в копирната машина няма листи за копиране метода createTracePaper

връща празен масив. Параметърът на метода `createTracePaper` се използва за това колко екземпляра да бъдат направени от копирания лист.

Задача 3

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

Създайте интерфейс `Move` с метод `moving`.

Реализирайте клас `животно (Animal)`, който описва енергията (`energy`) на животното до 100% абстрактно поведение за спане (`sleep`) и ядене (`eat`), като имплементира интерфейса `Move`.

Реализирайте клас `крава (Cow)`, който имплементира процеса за спане на кравата, като увеличава енергията и с 10% (`sleep`) и ядене (`eat`) като увеличава енергията и с 5% и имплементира метода `moving` с който се намаля енергията с 20%.

Реализирайте клас `кон (Horse)`, който имплементира процеса за спане на кравата, като увеличава енергията и с 7% (`sleep`) и ядене (`eat`) като увеличава енергията и с 15% и имплементира метода `moving` с който се намаля енергията с 15%.

Реализирайте клас `овца (Sheep)`, който имплементира процеса за спане на кравата, като увеличава енергията и с 5% (`sleep`) и ядене (`eat`) като увеличава енергията и с 10% и имплементира метода `moving` с който се намаля енергията с 25%.

Задача 4

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

Да се реализира клас `Персонаж (Character)`, който има име (`name`), живот (`life`) и абстрактно поведение нанасяне на удар (`striking`) връща като резултат стойността на удара и отбраняване (`defense`), приема като стойност нанесени щети.

`Character` се наследява от клас `Защитник (Defender)` и се разширява със защитно умение (`defensiveSkill`), броня (`armor`) и лечителско умение (`healingSkill`), живота му е 250 при създаване. Имплементира метода (`striking`), който връща като резултат стойността на удара 10. Метода `defense` поема като входен параметър нанесени щети, който се вадят от стойността на бронята, ако бронята е нула и има остатък от стойността на нанесените щети те се вадят от живота, след това се активира лечителското умение което носи по 1% възстановяване за всеки 100 единици но не повече от 100% от живота на персонажа.

Пример: живот: 250, броня: 150, лечителско умение: 350, нанесени щети: 400

Резултат: живот: 8,75, броня: 0, лечителско умение: 350

`Character` се наследява от клас `Атакуващ (Attacker)` и се разширява с ефективност (`efficiency`), живота му е 150 при създаване. Имплементира метода (`striking`), който връща като резултат стойността на ефективността. Метода `defense` поема като входен параметър нанесени щети, който се вадят от стойността на живота.

Задача 5

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

Да се реализира клас Фирма (Company), определящ се от: Име на фирмата (name), единен идентификационен код (EIK) – уникален 9-цифрен код, приходи за текущата година от дейността (incomes), разходи за текущата година от дейността (costs), методи за извличане стойностите на полетата. Да се създаде абстрактен метод, който връща като резултат дължимия данък върху печалбата от фирмата.

Класът Фирма да се наследява от:

- Фирма ООД (CompanyLtd), който я разширява с размера на капитала на фирмата (capital) и финансов резултат от предходна година (profit). Дължимият данък при тази правна форма се изчислява в следната последователност:

1. Изчислява се оперативната печалба, като от приходите от дейността се извадят разходите за дейността. Ако полученото число е отрицателно, данък не се дължи.

2. Ако финансовият резултат от предходната година е отрицателен, той се приспада от оперативната печалба. Ако полученото число е отрицателно, данък не се дължи.

3. Ако полученият резултат е положителен, данъкът се изчислява като 10% от стойността му.

- Фирма ЕТ (CompanyST), който разширява Фирма с име на собственика (owner) и поле, което указва дали упражняваната от него дейност е патентна или не (isPatentActivity).

Дължимият данък при ЕТ в следната последователност:

1. Ако ЕТ упражнява патентна дейност, данъкът е фиксиран за цялата година и е в размер на 500 лв.

2. Ако дейността не е патентна, данъкът е на стойност 15% от оперативната печалба, ако той е положителен.

Задача 6

Подберете подходящи модификатори за достъп в задачата и поставете сорс файловете в пакет `bg.tu_varna.sit.task6`

Да се реализира клас Building, който описва сграда с адрес (address) ширина (width), дължина (length) и височина (height). Да се добави абстрактен метод, за изчисление на отопляемия обем (heatedVolume) на сградата и разгъната площ (totalArea) на сградата.

Да се реализира клас House, разширяващ класа Building, който допълнително задава брой етажи и име на собственик. Площта (totalArea) се изчислява като площ на един етаж умножена по броя на етажите (приема се че всеки етаж има еднаква ширина, дължина и височина). След като се построи къщата, тя може да променя само името на собственика си. Отопляемият обем

на къщата се изчислява като площта на един етаж се умножи по височината на къщата и от резултата се приспадат 25% заради неотопляеми помещения и конструкцията на сградата.

Да се реализира клас Block, разширяващ класа Building, който допълнително задава брой входи (entrances), брой апартаменти (apartments) във всеки вход и метод за изчисляване площта на един апартамент (приемаме, че във всеки вход те са с еднакви дължина и ширина а височината им е 2,45). Отопляемият обем да се изчисли като произведение на стандартната височина на един апартамент (2.45), площта на един апартамент, броят апартаменти във вход и броя на входовете, редуциран с 10% заради неотопляеми помещения..

Задача 7

Подберете подходящи модификатори за достъп в задачата и поставете сорс файловете в пакет `bg.tu_varna.sit.task7`

Да се реализира клас Билет (Ticket), който поддържа информация за името на представлението (performanceName), цена (price) и абстрактно поведение за цената на билета (getPrice).

Билетът се надгражда от стандартен билет (StandardTicket), важащ за един зрител, допълвайки билета с поле за използван или неизползван (valid). Метода цена (getPrice) връща като резултат стойността на билета, добавена при неговото създаване.

Стандартният билет се разширява от билет с намаление (DiscountTicket) за студенти и пенсионери, като добавя поле за името на ползвателя (userName). Метода цена (getPrice) връща като резултат цената на стандартния билет намалена с 50%.

Съществува и Групов билет (GroupTicket), който важи максимум за 20 зрители (userCount) и има поле за броя зрители включени в този билет (currentUserCount). При създаването на обект от клас групов билет (GroupTicket) се подава единична цена за един зрител. Класът дефинира метод добавяне на нов зрител (addUser) в групата, като се достигнат максималния брой зрители метода не увеличава текущия брой зрители (currentUserCount). Метода цена (getPrice) връща като резултат цената на стандартния билет умножена по максималния брой зрители, след което се изважда произведението на включените в този билет (currentUserCount), зрители и намалението от 10%.

Задача 8

Подберете подходящи модификатори за достъп в задачата и поставете сорс файловете в пакет `bg.tu_varna.sit.task8`

Да се дефинира интерфейс (Salary) с поле за работните часове на ден (8) и метод calculate

Да се реализира клас DrivingLicense (шофьорска книжка) с номер на документа (number) и валидност в месеци (validity).

Да се реализира клас Human, който описва човек с три имена (firstname, secondname, thirdname), пол (gender) и възраст (age).

Класа Human се наследява от клас Employee с брой часове, които служителят работи на ден (worksHours) и заплата на час (rate). Класа имплементира интерфейса Salary като реализира метод calculate (), който връща заплата, която служителя получава за един ден.

Класа Employee се наследява от клас Driver, който разширява класа с шофьорска книжка (drivingLicense) и брой на курсовете за превозване на пътници, които извършва за един ден (courses). Класа имплементира интерфейса Salary като реализира метод calculate(), която се определя като към резултата от calculate() имплементацията на класа Employee (заплата му за един ден) се прибавят още 5%, ако за деня шофьорът е извършил повече от 10 курса за превозване на пътници.

Клас Employee се наследява от клас Engineer и се определя от брой на машините (number of machines), които е създал за един ден. Класа имплементира интерфейса Salary като реализира метод calculate(), (), която се определя като към резултата от calculate() имплементацията на класа Employee (заплата му за един ден) се прибавят още по 0.1% от заплата на час за всяка създадена машина за деня.

Задача 9

Подберете подходящи модификатори за достъп в задачата и поставете сорс файловете в пакет bg.tu_varna.sit.task9

Да се реализира клас Car, който се определя от марка (brand), модел (model), цвят (color) (16-но число), мощност (power), тип двигател (engineType), тип скоростна кутия (transmission), година на производство (yearOfManufacture), изминати километри (mileage). Когато се създава кола тя е без изминати километри и цвѳта и е (CCCCC). Класа дефинира метод за стартиране на двигател (startup), който връща истина двигателя може да се стартира винаги.

Класа Car се наследява от клас Truck, който го разширява с товарен капацитет (capacity) и изминато време от стартирането на двигателя (elapsedTime). Метода за стартиране на двигател (startup), връща истина ако двигателя може да се стартира, когато изминатото време от стартирането на двигателя е по-малко от 6 часа или лъжа, като изминатото време е повече от 6 часа.

Класа Car се наследява от клас Bus, който го разширява с брой пътнически места (seats) и изминати курсове (courses). Двигателя може да се стартира с метода за стартиране на свигател (startup), ако не е превишен лимита на курсовете. Средния брой курсове на ден са 10, ако остатъка от разликата на изминатите курсове и максималния брой на ден е по-малко от 5 се връща, че не е превишен лимита (false), ако е повече от 5, че е превишен лимита (true).

Задача 10

Да се реализират класове, описващи окръжност (Circle), конус (cone) и цилиндър (cylinder). Всеки един от класовете да съдържа по два конструктора – един с параметри и един по подразбиране, които инициализират стойностите на полетата. Дефинирайте интерфейси за изчисление лицето на фигурите (Area) с метод (calculate) и обема им (Volume) с метод (calculate). Общите свойства на фигурите да се изведат в общ родителски клас (Shape), от който да не могат да се създават обекти.