Започнат на	сряда, 7 юли 2021, 08:02
Състояние	Завършен
Приключен на	сряда, 7 юли 2021, 11:00
Изминало време	2 часа 57 мин.
Оценка	**6,95** от 12,00 (**58**%)

Въпрос **1**

Отговорен

6,95 от максимално 12,00 точки

В задачата можете да използвате наготово класа `std::vector` от `<vector>`.

Маймуната Зимбу решил, че е крайно време да започне сериозен бизнес. Навремето, когато пристигнал в България, докато го карали към зоопарка, шофьорът минал прекалено близо до няколко университета. Така докато стигнат, Зимбу вече бил собственик на няколко бакалавърски дипломи и на две магистърски. Когато пристигнали в зоопарка, оттам му казали, че е прекалено квалифициран и не го пуснали вътре. Така той се оказал в сложна ситуация -- трябвало сам да си изкарва прехраната.

Зимбу разгледал дипломите си и особено му харесала една, издадена от "Вселенския университет за висши науки и съвършенство във всички области" от Горно Нанадолнище. На нея пишело: "ИТ специалист!" (с удивителна). И понеже сега "компютрите са модерни", Зимбу решил да продава компютри. Веднага си регистрирал фирма (ЕТ "Стратиджик инфлуенсинг енд трендсетинг – Зимбу Маймуната") и започнал дейността си в един гараж. Но Зимбу не бил вчерашен. Още с пристигането си тук, той бил разбрал, че не иска да работи и му се струвало съвсем естествено да не прави нищо, а да получава по много. Затова той иска да автоматизира дейността си и някой да му напише програма (разбира се, безплатно – "*Тия програмисти какво само искат!*"). С нея хората сами ще могат да си направят конфигурация и да си я продадат сами на себе си. Вашата задача е да помогнете на Зимбу.

А) Компютрите които ще продава Зимбу се състоят от различни **Компоненти**. Един компонент има списък от неговите характеристики и техните количества. Всеки компонент е immutable обект (не може да се добавят ядра на вече произведен процесор).

Реализирайте клас `property`, с който да представите характеристика. Характеристиките също трябва да бъдат immutable обекти. Характеристиката трябва да има:

- име -- символен низ от тип `std::string`;
- количество -- цяло число от тип `int`;
- ценови множител -- число от тип `double`.

Реализирайте клас `component`, с който да представите компонент. Компонентът трябва да има:

- Списък с характеристики, които го описват. Няма ограничение за техния брой -- може да се произволно много.
- Член-функция, която по подадено име на характеристика, връща нейното количество. Помислете как да предупредите потребителя, ако той потърси за характеристика, която не се съдържа в компонента.
- Цена. Тя се смята като сумата от количеството на всяка характеристика умножена по неговия "ценови множител". За целта добавете в класа член-функция, която да връща цената.

Понеже Зимбу тепърва прохожда в бизнеса, неговият магазин има само 3 вида компоненти, но още другата седмица ще направи поръчка за още 7 вида! Видовете компоненти които са налични в магазина са:

- Процесор (`cpu`) с характеристики: брой ядра, честота в MHz
- Памет (`ram`) с характеристики: количество гигабайти, брой чипове, брой цветове с които свети в RGB
- Хард Диск (`hdd`) с характеристики: количество терабайти, скорост на четене в мегабайти, скорост на писане в мегабайти

Реализирайте класове-наследници на `component` за всеки от посочените горе компоненти.

Пример за компонент от тип Процесор:

- 4 ядра с "ценови множител" 120лв
- 5.3 гигахерца с "ценови множител" 50лв за GHz (т.е. 0,05 за MHz)

Общата цена е 4 * 120 + 5300 * 0.05 = 745лв

Б) Зимбу притежава **Магазин** в който има списък с всички компоненти, които се предлагат в него. Зимбу задава видовете компоненти и наличностите при стартиране на програмата.

За целта програмата ще получава, като аргумент от командния ред (`argc`/`argv`), път към текстов файл. Тя трябва да отваря и прочита от този файл списък от компоненти и веднага след това да го затваря.

Всеки компонент се описва на отделен ред във файла. Всеки такъв ред започва с ключова дума, която указва типа на компонента (`cpu`,`ram`,`hdd`). След нея, разделени с интервали следват характеристиките му.

Напишете клас `store`, който представя магазин. Класът трябва да има методи, с които в него да могат да се добавят и премахват компоненти, да се намира броят на съхранените в него компоненти и да се достъпват те.

Напишете клас `store_builder`, който може да прочете съдържанието на файл с компоненти, да построи магазин по него и да го върне.

В) Зимбу знае че продажбата на сглобени компютри (или както гордо ги нарича той -- **Конфигурации**) е много по изгодно от продажба на отделни компоненти.

Реализирайте клас `configuration`, който представя конфигурация. Една конфигурация има:

- Списък от компонентите, които съдържа. Добавете подходяща член-функция (или член-функции), с които да може да се достъпват те и да се намира техният брой.
- В една конфигурация не може да има два компонента от еднакъв тип. Използвайте RTTI (`typeid`), за да осигурите това.
- Цена, която се смята като сумата от компонентите в нея и след това се сложи 7% надценка (Зимбу е хитър търговец).

Освен това Зимбу знае че клиентите са капризни. Затова иска от програмата да може всеки клиент да създава **Изискване** за неговата конфигурация.

Реализирайте клас `requirement`, който представя изискване. Изискването също съдържа списък от компоненти. Те също трябва да могат да се достъпват и да се намира техният брой. В него също не може да има два компонента от еднакъв тип.

Считаме, че една конфигурация удовлетворява дадено изискване тогава и само тогава, когато:

1. съдържа компоненти от същите типове като тези в изискването (но не е ограничена само до тях, може да съдържа и повече).
2. характеристиките на компонентите в конфигурацията са същите или по-добри от тези в изискването. Например ако има изискване, в което се търси процесор с 5 ядра, конфигурацията няма да отговаря на него, ако няма процесор или има процесор само с 2 ядра.

Упътване: забележете, че между конфигурацията и изискването има много общи черти. Дали не може общата функционалност да се изнесе в общ клас?

Г) Зимбу не иска да оставя клиентите сами да си сглобяват конфигурации. За тази цел той иска да има клас **Конфигуратор** който ще може да създава тези конфигурации. Компонентите ще могат да бъдат предоставяни от магазина само на конфигуратора а не директно на клиентите. Конфигураторът трябва да може по даден Магазин и Изискване да създаде Конфигурация, която покрива Изискването или да извести потребителя. Няма изискване как точно да се удовлетвори изискването -- с по-евтини или по-скъпи компоненти, достатъчно е само то да се покрива.

Когато един компонент се добави в конфигурацията, той трябва да се премахва от магазина -- Зимбу иска да е изряден пред НАП и има голям страх да не би нещо да не излезе при евентуална ревизия.

Д) Напишете програма която удовлетворява желанията на Зимбу. Програмата да зарежда видовете компоненти и наличностите при стартиране. След това програмата да предоставя възможността на потребител да създава Изискване и с него да създава Конфигурация през Конфигуратора. Накрая конфигурацията трябва да се изведе на екрана, заедно с цената ѝ.

82134.cpp

Коментар: - обектите се пускат по копия
А) [1.25/2.25]
- calculate_price() трябва да бъде private
- характеристиките не се фиксират при наследниците
Б) [2.5/3]
- изтрива се динамична памет, която не се заделя в класа
В) [1/3]
- липсва сравнение между изискване и конфигурация
- ConfigurationRules трябва да има член-данна vector
Г) [1.7/2.5]
- написано е много нечетимо
- тук трябва да се помисли как да се подсигурим, че само конгураторът може да създава конфигурация
Д) [0.5/1.25]

◄ Разписание за допълнително контролно по ООП-практикум за студентите, които го пропуснаха поради съвпадащо контролно по ОС (2021-07-08)

Отиди на ...

Теоретичен изпит от 11:15 на 2021-07-07 ►