

Поща

Направете програма, която реализира работа с поща.

- Направете клас Person. Всеки човек има име и граждански номер (ЕГН) , който за удобство ще бъде символен низ (ЕГН-тата могат да започват с нули). За валидация проверявайте дали той е съставен само от цифри и е дълъг 9 символа (+1 за затапващ символ, стават 10).
- Направете клас Address. Адресът се характеризира с град, улица (може да бъде пропусната), блок (също може да бъде пропуснат, ако е къща, но тогава трябва да има улица), вход и етаж/апартамент.
- Направете клас Mail . Mail има получател (представител на Person) и адрес на получателя, както и изпращач и адрес на изпращача. Писмото освен това има и функция, която връща цената за изпращане getPrice(). Нека тя да връща реално число и да бъде чисто виртуална функция, заедно с чисто виртуална функция print().
- Направете два класа, наследяващи Mail - Letter и Package. Letter пази цена за писмото и флаг valuable(ценни книжа, например, булев флаг). Тя се изчислява при създаване на класа. Package съдържа в себе си тегло и цена на килограм, както и флаг fragile (bool). getPrice() съответно връща за Letter - $price * (1 + 0.3 * valuable)$, а за Package - $(price * (1 + 0.2 * fragile) * weight)$.
- Дефинирайте клас Mailman. Пощальонът наследява Person, но освен това пази в себе си хетерогенен масив от Mail (тоест Mail**). Вече знаете как става това - масивът е динамичен и се оразмерява както е нужно. Пощальонът обаче има и тегло за пратките, които може да носи (ако е с камион или пеш ще са различни тегла, например). Теглото на всяко писмо (Letter) е 0.01, а на пратка - самото нейното тегло. При добавяне на ново писмо/пратка направете така, че да се проверява текущото тегло на пратките на пощальона. Ако не може да се побере новата пратка/писмо се изписва съобщение на конзолата, в противен случай елементът се добавя в масива. Към класа на пощальона да бъде включена една функция, която сортира масива по даден критерии (с подадена функция), както и принтиране на съдържанието на масива му. Направете функция, която при подаден списък с адреси да доставя пощата на всеки адрес - изписва се на конзолата за всеки адрес доставките съответно с техните print функции, след което пратките се премахват от масива. Може да се направи и с оператори за сравнение (критериите са по ваша преценка, но трябва да са спазени във всички оператори - не може дадена пратка да е по-голяма и по-малка или равна от друга едновременно). Пощальонът трябва да има функция, която смята общата сума на товара си и функция, която връща екстремалност - най-евтина/най-скъпа/най-тежка доставка, например.

За проекта ще бъдат оценявани : правилно дефиниране на класове с всички спазени критерии, изучавани дотук. Класовете, които следва да имат наредби, също е

добре да имат оператори за сравнение. Ще се гледат изрично динамичните реализации на контейнера и при необходимост ще се правят допълнителни функции на място, ако някой от критериите не е спазен. **Проектът трябва да има работеща main функция, която показва пълната функционалност! При липса на такъв ще се пише на място!**