

Да се разпише **абстрактен** клас **Product** който съдържа свойствата **ID (код на стока каталоген номер), Name, Price, Quantity, Discount**. Свойства да са заключени за промяна. Класа да имплементира интерфейс **Comparable**. При сортиране да се подреждат първо по цена в намаляващ ред, после по цена в намаляващ ред и накрая по азбучен ред. Класа да имплементира интерфейс **ICost** съдържащ метода **GetCost** за изчисляване на общата стойност на продукта според количеството му в магазина.

Класа да съдържа публични методи

- **SetPrice** – за промяна на цената на стока
 - **SetQuantity** за промяна на наличното количество
 - **SetDiscount** – за задаване на отстъпката в проценти, първоначално да е 0
 - предефиниран **ToString** който вади данните за стоката в подходящ вид
 - виртуалния метод **GetCost** връщащ общата стойност за този продукт
-

Да се разпише клас **Food**, който наследява **Product** и съдържа свойства **ExpiryDate** (срок на годност), **Unit** (мерна единица – кг, бр...), **Quantity** (количество). Класа да е с интерфейс **Comparable**. При сортиране да се подреждат първо по срок на годност започвайки с тези, които изтичат най-скоро, после по количество. При добавянето на стока се въвежда спрямо текущата дата колко дни е срока на годност, а конструктора записва за **ExpiryDate** съответната крайна дата (текущата + броя зададени дни)

Класа да съдържа публични методи

- **SetExpiryDate** за промяна срока на годност
 - предефиниран **ToString** който вади данните за стоката в подходящ вид
 - предефиниран метод **GetCost** връщащ общата стойност за този артикул
-

Да се разпише клас **Garment**(дреха), който наследява **Product** и съдържа свойства **Type** (мъжка, дамска, детска), **Size** (30, 32, 34 ... 58, XS, S, M, L, XL, XXL ... - да се използва **Enum** за типа), **Material** (вълна, кожа, полиестер...). Класа да е с интерфейс **IComparable**. При сортиране да се подреждат първо мъжките, после дамските, накрая детските, след това по размер в нарастващ ред.

Класа да съдържа публични методи

- **SetExpiryDate** за промяна срока на годност
 - предефиниран **ToString** който вади данните за стоката в подходящ вид
 - предефиниран метод **GetCost** връщащ общата стойност за този артикул с допълнителни 5% отстъпка за детски дреки
-

Да се разпише клас **Shoes**, който наследява **Product** и съдържа свойства **Type** (маратонки, ботуши, обувки, чехли...), **Size** (между 18 и 54), **Color** (черен, кафяв...). Класа да е с интерфейс **IComparable**. При сортиране да се подреждат първо по размер в намаляващ ред, след това по тип.

Класа да съдържа публични методи

- **SetExpiryDate** за промяна срока на годност
 - предефиниран **ToString** който вади данните за стоката в подходящ вид
 - предефиниран метод **GetCost** връщащ общата стойност за този артикул с допълнителни 8% отстъпка за обувките с най-голям и най-малък номер
-

Да се разпише клас **Other**, който наследява **Product** и съдържа свойства **Type** (бяла техника, черна техника, прибори, перилни материали ...), **Manufacturer**(производител). Класа да е с интерфейс **IComparable**. При сортиране да се подреждат първо по тип, след това по производител.

Класа да съдържа публични методи

- предефиниран **ToString** който вади данните за стоката в подходящ вид
 - предефиниран метод **GetCost** връщащ общата стойност за този артикул с като при цена над 500лв и наличност над 30 добавя допълнителни 12% отстъпка
-

Да се разпише клас **СписъкСтоки** който съдържа списък на стоките в магазина.

Класа да съдържа публични методи

- **AddProduct** – за добавяне на стока (да хвърля грешка ако вече има стока с такова ID)
- **RemoveProduct** – за премахване на стока по ID
- предефиниран **ToString** всяка на отделен ред, започвайки с хранителните, след това дрехите, обувките и накрая останалите.
- **Search** – търсене по част от името
- **ListFood** – списък на хранителните стоки
- **ListShoes(номер)** – списък на обувките с този номер
- **ExpiryDiscount** – намалява с 15% цената на всички стоки с изтичащ срок на годност (оставащи до 5 дни)
- **ListExpensive(цена)** – стоките с цена над посочената (включително)
- **ListCheap(цена)** – стоките цена по-ниска от посочената (вкл.)
- **TotalCost** – обща сума на всички стоки в магазина (за целта да се ползва GetCost)

Да се разпише подходящ код в Main за тестване на класовете.