**Онлайн магазин**

**Общ преглед**

Във вашата фирма постъпва проект за създаване на онлайн магазин.

Вашият софтуер трябва да описва **продукти** (Product) и **поръчки (**Order**).**

Tрябва да реализирате функционалност, която да позволява добавяне на **продукти** и създаване на **поръчки**, включващи наличните продукти – всичко това ще работи чрез **команди**, които вие ще получавате. Поредицата от команди приключва с „**End**”.

За ваше удобство ще получите готов Program.cs файл, а вие ще трябва да реализирате само необходимите класове в Product.cs и Order.cs

**Подзадача 1: Структура на Продукт – 15 точки**

**Product**

Всички продукти имат име, цена и промоционален флаг:

name – низ, съставен от малки и големи латински букви, както и цифри, но без други специални знаци. Уникална стойност, която различава всички продукти

price – число с плаваща запетая

**isOnPromotion**– true / false

Реализирайте конструктори:

* **Product (string name, double price)** – този конструктор трябва да приема името и цената на продукта и да постави базисна стойност за промоционалното поле - false.
* **Product (string name, double price, bool isOnPromotion)** – този конструктор трябва да приема името на продукта, цената и промоционалния флаг.

За справка вижте следната схема за  Product.cs:

|  |
| --- |
| Product.cs |
| public Product(string name, double price)  {      //TODO: Добавете вашия код тук …  }    public Product (string name, double price, bool isOnPromotion)  {     //TODO: Добавете вашия код тук …  } |

**Команди за добавяне на продукти**

Трябва да реализирате следните две команди за добавяне на продукти:

* **CreateProduct** <**име**> <**цена**> - тази команда има за цел да добави продукт с неговото име, цена и базисна стойност за промоционалното поле - false
* **CreateProduct** <**име**> <**цена**> <**isOnPromotion**>- тази команда има за цел да добави продукт с неговото име, цена и стойност за промоционалното поле

**Команда за извеждане на информация**

Вашето приложение във всеки един момент може да получи заявка да отпечата информация за продукт. Командата за това е следната:

* **PrintProductInfoByName**<**име**> - отпечатва информация за продукт във формат:

***Product ->****<****име****> with price <****цена****>. On promotion <YES/NO>*

Цената, следва да бъде форматирана до втория знак след десетичния разделител. Ако продуктът е промоционален в края на текста трябва да се постави YES, в обратен случай – NO. Виж нулевия тест

Тази команда ще получава винаги валидни и съществуващи имена на продукти. За успешна реализация трябва да реализирате ваша версия на **ToString()**метода за класа **Product.**

**Подзадача 2: Структура на Поръчка – 15 точки**

**Order**

Всяка поръчка има: номер и списък от продукти.

**orderNumber**– цяло число

products – списък от **Product**

**Част от логиката на приложението е да се пази в променлива общия брой на всички поръчки. За целта трябва да бъде добавен брояч, който да бъде увеличаван с 1 за всяка поръчка.**

Трябва да реализирате конструктор:

* **Order(int**orderNumber**, List<Product>**productsList**)** – този конструктор трябва да приема номер на поръчката и продуктите, които са добавени към нея.

За справка вижте следната схема за  Order.cs:

|  |
| --- |
| Order.cs |
| public Order(int orderNumber, List<Product> productsList)  {      //TODO: Добавете вашия код тук … } |

**Команда за създаване на поръчка**

* **CreateOrder**<**orderNumber**> <ProductName> <ProductName>… - тази команда има за цел да създаде поръчка като в нея трябва да се добавят всички изброени продукти.

**Команда за извеждане на информация**

Вашето приложение във всеки един момент може да получи заявка да отпечата информация за поръчка. Командата за това е следната:

**PrintOrderByNumber < orderNumber > -**отпечатва информация за поръчка във формат:

Order #333333 has the following products:

### Product -> IphoneX with price 1190.92. On promotion: YES

### Product -> AirPods with price 149.90. On promotion: NO

…

Тази команда ще получава винаги валидни и съществуващи номера на поръчки. За успешна реализация трябва да реализирате ваша версия на **ToString()**метода за класа **Order.**Oчаква се преди всеки продукт да се поставят ###, а след това да бъде преизползван ToString() на съответния продукт

**Подзадача 3: Логика – 40 точки**

**Kоманди**

Вашето приложение трябва да реализира следните команди:

* **AddToОrder**<**orderNumber**> <**productName>**<**productName>…** -Тази команда добавя в дадената поръчка всички описани продукти
* **GetOrderTotalPrice**<**orderNumber**> **-**Тази команда трябва да принтира сумата на цените на всички продукти
* **GetOrderDiscountedProductsPrice**<**orderNumber** > - Тази команда трябва да принтира сумата на цените на всички **ПРОМОЦИОНАЛНИ** продукти
* **PrintProductInfoByName**<**productName**> - Изпечатване на продукта във формата описан по-горе
* **PrintOrderByNumber**<**orderNumber** > - Изпечатване на поръчката във формата описан по-горе

**Подзадача 4: Бонус логика – 20 точки**

Трябва да реализирате още няколко команди:

* **GetOrderDiscountedProductsCount**<**orderNumber** > **-**Следва да принтирате броя на промоционалните продукти налични в поръчката
* **GetTotalOrdersCount** **–**Изпечатва общия брой на всички създадени поръчки

*За безпроблемната работа на всички изброени команди от 3 и 4 подзадача трябва да реализирате Product.cs и Order.cs, по аналогичен начин на показаното по-долу:*

|  |
| --- |
| Order.cs |
| public Order(int orderNumber, List<Product> productsList)  {       //TODO: Добавете вашия код тук …  }    public static int OrdersCount  {       //TODO: Добавете вашия код тук …  }    public int OrderNumber  {       //TODO: Добавете вашия код тук …  }    public void AddProduct(Product product)  {       //TODO: Добавете вашия код тук …  }    public double GetOrderTotalPrice()  {       //TODO: Добавете вашия код тук …  }    public double GetDiscountedProductsTotalPrice()  {       //TODO: Добавете вашия код тук …  }    public double GetDiscountedProductsCount()  {       //TODO: Добавете вашия код тук …  }    public override string ToString()  {       //TODO: Добавете вашия код тук …  } |

|  |
| --- |
| Product.cs |
| public Product(string name, double price)  {      //TODO: Добавете вашия код тук …  }    public Product (string name, double price, bool isOnPromotion)  {     //TODO: Добавете вашия код тук …  }    public string Name  {       //TODO: Добавете вашия код тук …  }    public double Price  {       //TODO: Добавете вашия код тук …  }    public bool IsOnPromotion  {       //TODO: Добавете вашия код тук …  }    public override string ToString()  {     //TODO: Добавете вашия код тук …  } |

**Забележка:**Освен горепосочените методи трябва да реализирате и необходимите свойства за всеки от класовете. Възможно е да е удачно да реализирате допълнителни полета, свойства и методи, по ваша преценка.

**Подзадача 5: Валидация – 10 точки**

Освен всичко останало вие трябва да направите и валидация!

Не допускайте създаването на:

* Продукт с цена под 0.00, message => Price should be positive!
* Продукт с име по-късо от 3 символа, message => Invalid product name!

При невалидни данни хвърлете грешка от тип **ArgumentException(“message”)**

**Вход / Изход**

**Вход**

* Програмата ще получава множество редове с информация. Всеки ред представлява команда. Самият вход се обработва изцяло от примерния Program.cs.
* Всички команди приключват с въвеждането на End

**Изход**

За някои от командите не е нужно да извеждате нищо. За всички останали изпечатването ще ви бъде дадено в Program.cs, освен ToString() методите, които са оставени на вас.

**Ограничения**

* Всички цели числа ще бъдат в диапазона **–10000** до **+10000**
* Всички числа с плаваща запетая ще бъдат въвеждани с до два знака след запетаята.
* Всички имена няма да съдържат интервал

**Примери**

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| CreateProduct AirPods 149.90  CreateProduct IphoneX 149.90 true  CreateProduct IphoneXR 149.90 false  CreateOrder 333333 IphoneX  AddToOrder 333333 AirPods  GetOrderTotalPrice 333333  GetOrderDiscountedProductsPrice 333333  GetOrderDiscountedProductsCount 333333  PrintProductInfoByName IphoneX  PrintOrderByNumber 333333  GetTotalOrdersCount  End | 269.82  119.92  1  Product -> IphoneX with price 119.92. On promotion: YES  Order #333333 has the following products:  ### Product -> IphoneX with price 119.92. On promotion: YES  ### Product -> AirPods with price 149.90. On promotion: NO  1 |

**Точки**

Разбивката по подзадачи е следната:

1. **15** точки, като трябва да имате и задължително реализиран ToString() метод
2. **15** точки, като трябва да имате и задължително реализиран ToString() метод
3. **40** точки
4. **20** точки
5. **10** точки

Общ брой точки: **100**

Скала за оценяване:

0-15т – Слаб 2

15-30т - Среден 3

31-45т - Добър 4

46-60т - Мн. добър 5

над 61т - Отличен 6