

**Софийски Университет “Климент Охридски”**  
**Факултет по Математика и Информатика**

**Контролно No. 1a**

**Курс:** Обектно Ориентирано Програмиране с C#.NET

**Преподавател:** д-р. Е. Кръстев

**Студент :**

**Дата:** януари 6, 2021

**Време за работа:** 120 min

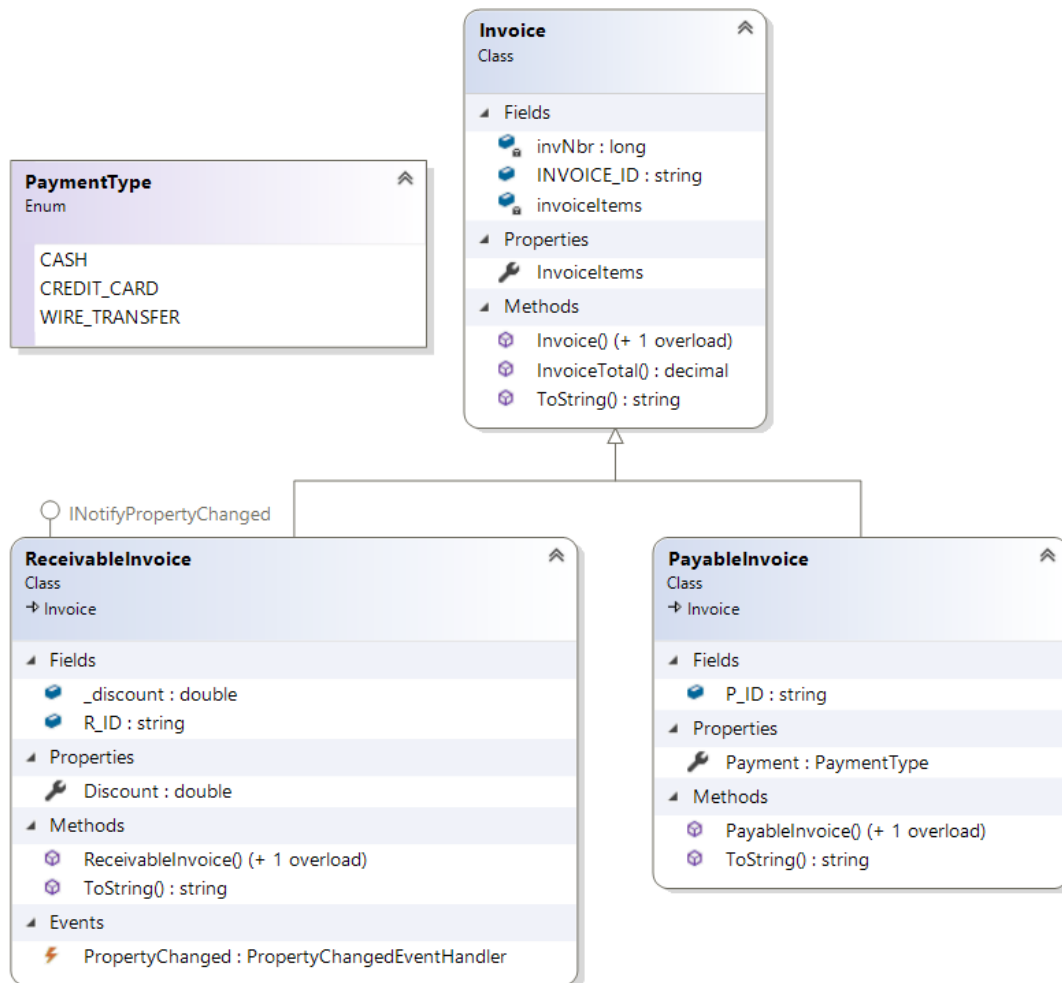
**Инструкции:** Изпълнете следното задание за обектно ориентирано програмиране и предайте пълния набор от файлове необходими за решаване на програмата на флопи диск. Пълен набор от точки се присъжда за пълно решение на съответната подзадача.

**Оценки:**

2	от 0 до 54 точки
3	от 55 до 64 точки
4	от 65 до 74 точки
5	от 75 до 84 точки
6	от 85 до 100 точки

**Задача 1 ( 100 точки)**

Създайте библиотека (Class Library) от класове, която да използвате в **WPF .NET Framework** приложение за обработка на фактури. Библиотеката съдържа класове със следната UML диаграма:



Изпълнете този ОО модел в следната последователност: (Създайте потребителските типове данни в отделни файлове като ползвате **означенията, въведени по-долу**)

1. **Напишете дефиницията** на `enum PaymentType`. Той задава следните типове плащания. `CASH`, `CREDIT_CARD`, `WIRE_TRANSFER`

Точки: 2

2. **Напишете дефиницията** за `class Invoice`. Направете **всеки** `Invoice` обект да **има** (уникален) публично достъпен **пореден** `string` номер `INVOICE_ID` (константа за съответния обект), имащ **префикс** "I" и следван от **6-** цифрено число, чиито **незначещи цифри са заменени с нули..** Всяка инстанция на класа **има** **масив** `invoiceItems` от данни за фактурираните продукти, където **типът на елементите на масива е** `ValueTuple`, съдържащ **описание на продукта** и **неговата цена** съответно посредством `string description` и `decimal cost`.

Точки:6

3. **Напишете** в `class Invoice`
  - `get` и `set` property за `invoiceItems`
  - **Дефинирайте** конструктори "за **общо ползване**" и по **подразбиране** за `class Invoice`, където се инициализират данните на класа- `INVOICE_ID` и `invoiceItems`.
  - Метод `public decimal InvoiceTotal()`  
който връща **резултата от изпълнение на LINQ заявка** за сумиране на цените `cost` в масива `invoiceItems`
  - Предефинирайте метода `ToString()`, наследен от клас `object`, така че да връща стойността от изпълнението на `InvoiceTotal()`, **форматирана** като валута и **предхождана** от текста "`Invoice total :`" (вижте примера по- долу)

Точки:10

4. **Напишете** следните **производни класове** на `class Invoice`:
  - `class ReceivableInvoice`
  - `class PayableInvoice`**Всеки** обект на `ReceivableInvoice` и `PayableInvoice` да **има** (уникален) **публично достъпен пореден** `string` номер съответно `R_ID` и `P_ID` (константи за съответния обект). Тези константи се инициализират от константата `INVOICE_ID` към която се добавят съответно **префикси** "R" и "P".  
Отделно, `class ReceivableInvoice` има **пълно C# свойство** `Discount` (задава процент на отстъпка при пресмятане на `ReceivableInvoice` фактура), а `class PayableInvoice` има **автоматично имплементирано C# свойство** `Payment` от тип `PaymentType`.

Точки:10

5. **Напишете** във всеки от тези класове:

конструктори “за **общо ползване**” и по **подразбиране**

- Предефиниране на метода ***ToString()***, наследен от клас ***object***, да връща **номера на фактурата** на текущата инстанция, **размера на отстъпката**(***Discount***, **число в интервала [0.00, 1.00)** или съответно **начина за плащане**(***Payment***), следвани от **форматираната сума на цените *cost* в масива *invoiceItems***.

Точки:8

6. **Публикувайте** събитие ***PropertyChanged*** в ***class ReceivableInvoice*** и **обработете** това събитие при промяна на свойството ***Discount***

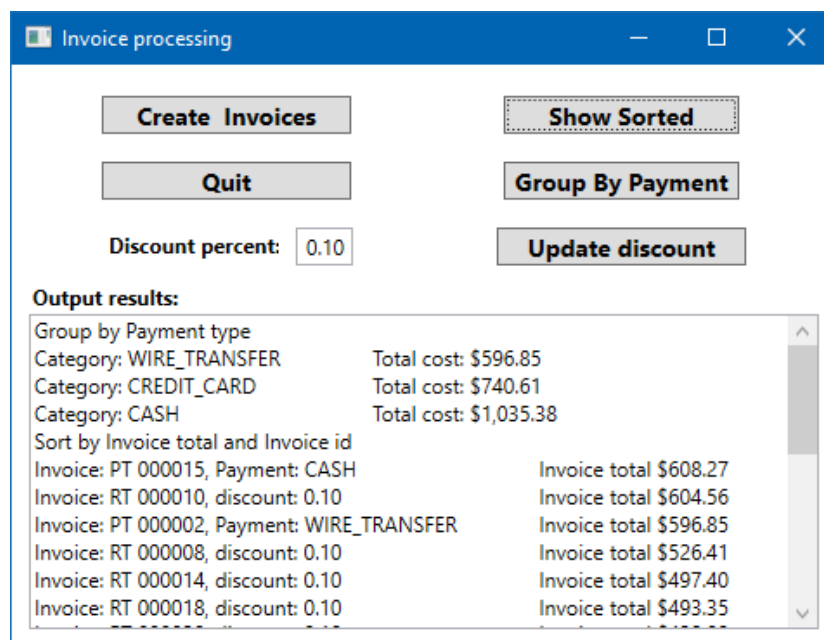
Точки:4

7. **Създайте WPF .NET Framework** приложение и **реферирайте в него потребителската библиотека** с типовете данни в зададената UML диаграма на класове

Точки:5

8. **Създайте** следния потребителски интерфейс в **WPF** приложението като използвате **подходяща логическа структура** от панели, съдържание и стилове за възпроизвеждането му. **Задължително** използвайте **смислени имена за идентификатори на контроли** и **методи за обработка на събития**, посредством **префикси** за означаване на типа на контролите по Изменената Унгарска Нотация по примерите в лекциите.

Точки: 22



9. **Добавете** списък ***invoices*** от ***Invoice*** елементи в едноименния клас, съпътстващ XAML описанието на потребителския интерфейс и **напишете следните методи** за обработка на събития:.

- a) **Напишете метод *ManageDiscounts*** за обработка на събитие ***PropertyChanged***, който намалява всички цените (*cost*) на всички продукти **на** елементите от *invoices*, които са от тип ***ReceivableInvoice*** с текущата стойност на свойството *Discount* на тези обекти (5 точки)
- b) При натискане на бутона **Create Invoices** се генерират посредством генератор на случайни числа данни за 20 *ReceivableInvoice* и *PayableInvoice* обекта, които се добавят към списъка *invoices*. Инициализирайте тези обекти в цикъл. Във всяка от итерациите на цикъла: (12 точки)
- По случаен начин да се определи типът *ReceivableInvoice* или *PayableInvoice*
  - Дължината на масива *invoiceItems* да е число случайно избрано в интервала [1, 10]
  - Поредният елемент на масива да има за описание (*description*) "Item " и суфикс пореден номер в масива, а цената (*cost*) на покупката да е произволно избрано числа в плаваща запетая в интервала [100.00, 200.00]
  - Типът за плащане *PaymentType* на *PayableInvoice* фактура да се определи с генератора на случайни числа
  - Размерът на отстъпката(*Discount*) на *ReceivableInvoice* да се прочете от текстовото поле *Discount percent*
  - След добавяне на обект *ReceivableInvoice* към списъка *invoices* да се извърши абониране за събитието *PropertyChanged* на този обект посредством метода *ManageDiscounts*
- c) При натискане на бутона **Quit** да се прекрати изпълнението на програмата
- d) При натискане на бутона **Update discount** да се зададе нова стойност на свойството *Discount* на всички елементи от списъка *invoices*, които са от тип *ReceivableInvoice* (5 точки)
- e) При натискане на бутона **Show Sorted** да се изведат в **Output results** елементите на *invoices*, сортирани низходящо с LINQ по *InvoiceTotal()* (5 точки)
- f) При натискане на бутона **Group by Payment** да се изведат в **Output results** *PayableInvoice* елементите на *invoices*, групирани с LINQ по типа на плащане (*Payment*) (5 точки)