Софийски Университет "Климент Охридски"
Факултет по Математика и Информатика

Контролно No. 1

Курс: **Обектно Ориентирано Програмиране с C#.NET**
Преподавател: д-р. Е. Кръстев **Студент** :
Дата: декември **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)

Приложете следните принципи на Обектно ориентираното програмиране:

- ***hiding of information***
- ***software reuse***
- ***inheritance***
- ***polymorphism***

Един ИТ специалист, работещ в компания за счетоводни услуги, трябва да **обработва** получаваните **фактури за приходи (***Receivable***) и разходи (***Outgoing***).** За тази цел той използва следния **Object oriented model** в **C#.NET.**

Дефинира ***`class InvoiceDetail`,*** за да моделира ***отделните стоки*** **,** описани в **дадена *фактура*.** За да разграничи операциите по приходните от разходните фактури той въвежда ***`interface IReceivable`*** и ***`interface IOutgoing`*** в ОО модела, за да моделира **разходните фактури като** всяка от тези фактури има описание за **закупена**/ **платена** една **или** повече стоки (***`InvoiceDetail`*** objects). За удобство, ***предполагаме,*** че знаем **броя на тези стоки** по време на създаване на дадена фактура. Допълнително изискване е, да **сумираме общата сума по фактурите и да добавяме масив от новозакупени стоки към съществуваща фактура**.

Изпълнете така описания ОО модел **като C#.NET Console application в следната последователност**: *(Използвайте **отделни файлове** за всеки `class` или `interface`. Използвайте **нотацията за** classes, methods и objects , **въведени по-долу**)*

1. Да се **създаде** проект от тип ***Class Library*** , **включващ** по- долу описаните ***`class Invoice, class InvoiceDetail,`*** както и интерфейсите ***`IReceivable и IOutgoing`***. **Именувайте** този проект ***`Utilities`***.

Точки: 2

2. **Напишете дефиницията** на ***interface IReceivable*** и ***interface IOutgoing.*** ***Всеки*** от тях **има** ***get*** **property**, именувано ***InvoiceTotal***, за получаване на **общата парична стойност** на съответната фактура.

Точки: 4

3. **Напишете дефиницията** на **class** ***InvoiceDetail.*** Всеки ***InvoiceDetail*** обект **има** един ***decimal dblLineTotal*** class member. **Дефинирайте** конструктор "***за общо ползване***" и **онаследеният метод** ***ToString***() от ***class object***, който **връща форматиран стринг** *(парична сума с два знака след десетичната запетая)* на ***dblLineTotal*** на текущия обект.**.** **Дефинирайте** също ***get*** и ***set*** **property** за ***dblLineTotal*** class member.

Точки: 8

4. **Напишете дефиницията** за **class** ***Invoice.*** Нека **всеки** ***Invoice*** обект да има (**уникален**) **пореден** ***long*** номер, който е **константа**.. Нека този номер се **представя** от данната ***INVOICE_NUMBER***. Допълнително, нека обектите на **class** ***Invoice*** да **имат масив** от ***InvoiceDetail*** елементи***.*** ***Именувайте*** този масив като ***invoiceItems***. **Дефинирайте** **конструктор** "***за общо ползване***" за **class** ***Invoice***, където се инициализират данните на класа и онаследеният метод ***ToString***() от ***class object***, който **връща форматиран стринг** с всички данни на текущия обект.

Точки:16

5. **Напишете** ***get*** **property** за данната ***INVOICE_NUMBER*** , а също ***get*** и ***set*** **property** за ***invoiceItems.***

Точки:10

6. **Напишете** метод ***InvoiceTotal*** в **class** ***Invoice,*** който **връща сумата** от ***dblLineTotal*** за всички ***InvoiceDetail*** елементи на масива ***invoiceItems.***

Точки:6

7. В ***class Invoice*** **предефинирайте** метода ***Equals***() , **онаследен** от ***class object***, така че **две фактури да са равни**, когато съответния им **метод** ***InvoiceTotal*** **връща еднакви стойности**

Точки:10

8. **Имплементирайте с явно цитиране на името** ***IReceivable*** и ***IOutgoing*** **в class** ***Invoice***, така че **свойството** ***InvoiceTotal*** спрямо интерфейс ***IReceivable*** **връща** ***InvoiceTotal***()), спрямо интерфейс ***IOutgoing*** **връща** ***InvoiceTotal***() с **обратен знак**.

Точки:10

9. В ***class Invoice*** напишете **статичен** метод, именуван `PrintInvoices`. Този метод има един аргумент- **списък** от ***Invoice*** обекти и **за всеки** елемент от списъка,**отпечатва** на ***Console*** съответно ***INVOICE_NUMBER*** , след това на отделни редове ***DblLineTotal*** стойностите за всяка от ***invoiceItems***, **сортирани в низходящ ред** на ***DblLineTotal*** Накрая **отпечатва и стойността на свойството** ***InvoiceTotal*** в зависимост от елемента от списъка е ***IReceivable*** или ***IOutgoing.*** Всички **парични суми трябва** да се **форматират като *currency*** с **два знака** след десетичната запетая и **да се използва LINQ за сортиране**.

Точки:10

10. Напишете **разширяващ** метод `AddAllInvoice` за ***class Invoice***, който добавя елементите на **масив** от `InvoiceDetail` обекти(подаден като **аргумент** на метода) **към масива** ***invoiceItems*** на ***class Invoice.***

Точки:10

11. **Създайте нов проект от тип** конзолно приложение **в същия Soluiton. Добавете библиотеката** ***Utilities*** и **напишете следните команди** в метода ***Main()*** на ***class Program***:.
    - **Използвайте генератор на случайни числа** и **създайте масив** `details` от 10 ***InvoiceDetail*** **обекта**, чиито ***dblLineTotal*** са в **интервала** [0, 50].
    - команди **за създаване** на ***IOutgoing*** фактура ***in1*** с ***invoiceItems***= {***details[0]***} и ***IReceivable*** фактура ***in2*** с ***invoiceItems***={***details[1]***}
    - **изпълнете разширяващия метод** `AddAllInvoice` на ***in1*** и **добавете** масива `details` **към фактурата** ***in1***
    - команди **за създаване списък** (именуван ***myInvoices***) от ***Invoice*** обектите ***in1*** и ***in2.***
    - **Изпълнете метода** `PrintInvoices` **на *class Invoice* със списъка** от фактури ***myInvoices***
    - **Сравнете** ***Invoice*** обект ***in1*** със ***in2*** като **използвате *Equals***() метода и **изведете** резултата от сравнението, както и текстовото описание на ***in1*** и ***in2*** с ***ToString()*** методите им

Точки:14