

**СОФИЙСКИ УНИВЕРСИТЕТ “КЛ. ОХРИДСКИ”
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА**

Писмен изпит No. 1a

Предмет: Обектно ориентирано програмиране на C#.NET

Преподавател: д-р. Е. Кръстев

Студент :

Дата: февруари 19, 2022

Време: 120 min

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 на C#.NET:

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

при намиране решението на следната задача за получаване на обменния курс на валута от стокова борса, чрез заявки към отдалечен обект на сървера на борсата. Обменният курс се извежда в панели на *MS Windows* приложения, които четат непрекъснато променящия се курс на стоковата борса, чрез заявка за услуга към отдалечен обект на сървер на стоковата борса.

Решете поставената задача в описаната по- долу последователност:

1. Напишете *Class Library* проект *StockExchange*. Дефинирайте изброим тип *Money*, който съдържа следните константи *EUR_USD*, *GBP_USD*, *USD_JPY*, *USD_RUB*

3 точки

2. Добавете *class Currency* в проекта *StockExchange*. Той има
 - данна *currencyCode* от тип *Money*
 - данна *exchange- decimal* стойност на обменния курс,
 - статична данна *rand- Random* генератор на случайни числа (по подразбиране инициализирана)

Напишете:

- *set* и *get* свойства за нестатичните данни на класа
- конструктори за общо ползване, по подразбиране (код *EUR_USD*, курс 1.13) и конструктор за копиране
- предефинирайте метода *ToString()*, който да извежда *currencyCode*, обменния курс *exchange* с 4 знака след десетичната запетая, както и текущото време във формата *hh:mm:ss*, чрез съответните свойства на *DateTime.Now*.

12 точки

3. Добавете в проекта *StockExchange* дефиницията на интерфейс *ITradeService*, който има *get* свойство *Trade* от тип *Currency* едномерен масив .

2 точки

4. Нека *class Currency* има метод

```
void ChangeExchangeRate ()
```

който променя обменния курс `exchange` по случаен начин с използване на генератор за случайни числа в затворен интервал, образуван от **+/- 3%** изменение от текущата стойност на `exchange`. Да се вземе предвид, че обменният курс се отчита с не повече от **4 знака** след десетичната запетая.

4 точки

5. Добавете в проекта *StockExchange* class *TradeProduct*, който е *Serializable*. Нека *TradeProduct* има

- данна `moneyToTrade` от тип *Currency* масив
- `get` и `set` *thread-safe* свойства за `moneyToTrade`
- Инициализирайте референцията `moneyToTrade` в конструктора по подразбиране на class *TradeProduct* като масив от четири *Currency* елемента със следните данни

Код валута	0	1	2	3
Обменен курс	1.130	1.65	107.102	34.37

8 точки

6. Нека class *TradeProduct* имплементира интерфейс *ITradeService* с явно цитиране на името му, при което свойството *Trade* връща текущите стойности на масива `moneyToTrade`

4 точки

7. Напишете в проекта *StockExchange* също и дефиницията на потребителски WPF контрол *StockExchangeControl* има многоредово текстово поле и също:

- референция към масив `currency` от тип *Currency*
- делегат *Traders* за `void` методи, които приемат по референция без предварителна инициализация масив от *Currency* елементи
- събитие *Broker* от типа на делегата *Traders*

Нека след инициализиране на *StockExchangeControl* да се стартира нишка, която изпълнява метод `RunStockExchange`.

10 точки

8. В class *StockExchangeControl*

- a. Напишете метод

```
void RunStockExchange ()
```

който в бекраен цикъл да:

- изстрелва събитието *Broker* като за аргумент използва масива `currency`
- „приспива” текущата нишка, изпълняваща метода, за 1000 ms
- изписва в текстовото поле на *StockExchangeControl* текущите стойности за елементите на масива `currency` като използвате `ToString()` метода на class *Currency*

- b. Напишете конструктор по подразбиране на class

StockExchange в който да се създаде и стартира нишка за изпълнение на `RunStockExchange`

10 точки

9. Напишете нов проект, WPF приложение *StockClientApp*, Нека този проект съдържа WPF прозорец *StockClient* и служи за показване на обменния курс на борсата в клиентското приложение след промяната му от сървера

на борсата. посредством визуалната контрола `StockExchangeControl`. Добавете данна в съпътстващия файл `ITradeService trade` която инициализирайте на обект, прочетен с десериализиране от порт 12345 на localhost. Използвайте `TcpClient` за осъществяване на мрежовата връзка.

17 точки

10. В XAML описанието на `StockClient` „закачете“ събитието `Broker` метод `void UpdateCurrency(out Currency[] trade)` за обработка на това събитие. Нека реализацията на метода `UpdateCurrency` да инициализира аргумента `trade` посредством свойството `Trade` на `StockExchangeControl`

10 точки

11. Напишете нов проект, конзолно приложение `StockExchangeServer` където в `main()` метода да се създава и регистрира `TcpListener` за сериализиране на обект от клас `TradeProduct`. Нека достъпът да се осъществява на порт 12345 от localhost където с многонишков достъп на множество от TCP клиенти да се изпраща сериализирано един и същи обект от клас `TradeProduct`

20 точки

