

**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 5, 2024

**Време:** 120 min

**Инструкции:** Отбележете отговорите на теоретичните въпроси на този лист, а пълния набор от файлове необходими за решаване на програмата запишете на флопи диск.

**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 клиент-сървър приложение на .NET за обслужване на поръчки за провеждане на образователни курсове, където поръчките създават от клиента, а се обработват и съхраняват на сървъра. Отделните клиенти правят поръчки в графични прозорци на WPF приложения, които позволяват да се заяви курс за конкретен брой обучавани. При заявяване на курс за брой на обучавани, надхвърлящ, зададена прагова стойност, сървърът генерира допълнителни курсове от същата тематика, но с брой обучавани в рамките на праговата стойност за максимален брой участници в курса. Поръчките на клиентите се записват в текстов файл на сървъра на фирмата организираща курсовете.

Решете поставената задача в описаната по- долу последователност:

A. Създайте на WPF (.NET) проект *EducationServicesLib* с Output type *Class Library*, имащ следното съдържание

1. Дефинирайте в проекта *EducationServices* изброим тип *ServiceType*, който съдържа следните константи  
ONLINE, CLASS\_ROOM, INTENSIVE, ON\_DEMAND, HIGH\_LEVEL

**2 точки**

2. Добавете *struct Course* в проекта *EducationServicesLib*. Нека *Course* има свой публично достъпен (уникален) пореден *string* номер (*string* константа за съответния обект), който започва с четири-цифрено число, чиито незначещи цифри са запълнени с нули и завършва с “-”, следвано от числото на текущата година (примерно, 2024) на генериране на обекта *Course*. Нека този номер се представя от data member именуван *ID* и се инициализира в конструкторите на class *Course*.

Публична статична константа *MAX\_IN\_COURSE*, задава максимален брой обучавани в курс, по подразбиране е 10.

Нека *Course* обекта има още:

- данна *serviceType* от тип *ServiceType*
- данна *title* наименование на курс
- данна *numOfStudents*– брой обучавани ненадхвърлящ *MAX\_IN\_COURSE*

Напишете:

- `set` и `get` свойства за всички `private` данни
- конструктор за общо ползване и конструктор за копиране
- предефинирайте метода `ToString()`, който да извежда ID и всички останали данни на обекта, всички подходящо форматирани.

12 точки

3. Създайте нов (.NET) проект (към същия Solution), `EducationServer` с Output type Console Application за реализация на многонишков TCP сървър. Добавете към този проект референция към потребителската библиотека `EducationServicesLib`.

Добавете към `class Program`

- данна `courses` от тип `Dictionary<string, Course>` таблица от кода ID на курс и курса, съответстващ на кода
- данна `titles` от тип `List<string>`, съдържащ заглавия на предлагани курсове
- данни

Инициализирайте `courses` и `titles` в конструктора по подразбиране на `class Program` като изберете с генератор на случайни числа:

- Първоначалният брой на елементите на `titles` и `courses` е еднакъв, произволно число в интервала [6, 20]
- Елементите на `titles` да се образуват от "Title No. ", следвано от поредния номер на елемента
- Стойностите на `courses` да се създават като обекти на `class Course` така
  - Броят на елементите на `courses` е произволно избран в [6, 20]
  - Типът `serviceType` на всеки курс е произволно избран `ServiceType`
  - Заглавието `title` е пореден елемент от `titles`
  - Брой на обучавани в курса да е произволно в интервала [6, 20]

16 точки

4. Добавете към `class Program` метод

`void RunServer()`

Нека този метод да създава и стартира `TCPListener` и в безкраен цикъл да обслужва TCP клиенти на IP 127.0.0.1 и номер на порт по избор на студента.

Добавете също метод

`void ProcessOrder(object socket)`

който да се изпълнява в нишка от `ThreadPool`, която да се стартира след установяване на връзка с клиента.

Добавете прихващане на изключения и затваряне на ресурсите, които съответно възникват и се използват за комуникация с клиентите.

6 точки

5. Реализирайте следната функционалност в метода `ProcessOrder`

- Създават се инстанции на `NetworkStream`, `BinaryReader` и `BinaryWriter` съответстващи на създадената връзка с клиент
- Изпращат на текущо обработвания клиент:  
`string` с елементите на списъка `titles`, разделени със запетаи  
`string` с константите на `ServiceType`, разделени със запетаи
- В безкраен цикъл изпълнява команда на `BinaryReader` за прочитане на `string`, съдържащ данни за заявените курсове от клиента. След прочитане на данните, те се обработват в метода `void WriteData(string data)`, описан по-долу.

Добавете прихващане на изключения и затваряне на ресурсите, които съответно възникват и се използват за комуникация с клиента.

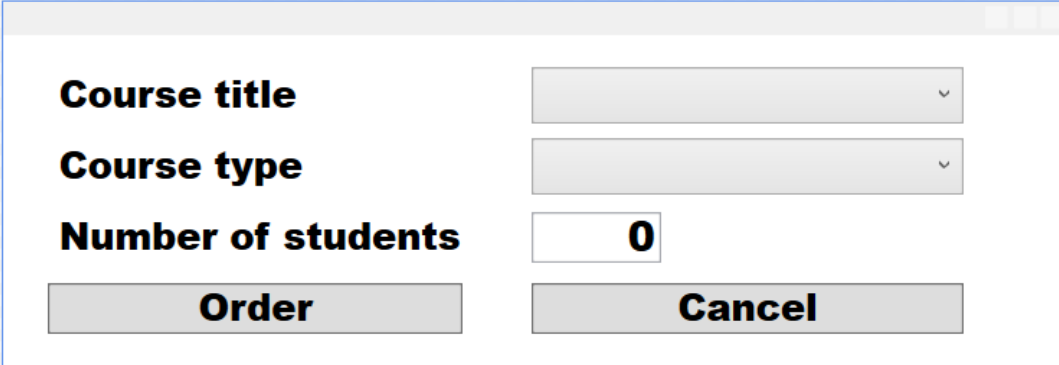
8 точки

6. Методът `WriteData` записва в текстов файл данни за заявки за `Course` от клиент, представени в параметъра на метода, където тези данни са записани поотделно в `string` и са разделени със `'\n'`. Типът и подреждането на тези данни е описан в клиентското приложение (виж т. 9). Поради това, първо извличете отделните данни от този ги използвайте за създаване на обект `course` тип `Course`. След това добавете `course` в `courses` и запишете **thread-safe** в текстов файл, именуван по избор на студента, ред с текстовото представяне на създадения обект от тип `Course`. Ако `course` има брой на обучавани `numOfStudents > MAX_IN_COURSE`, броят на обучавани в този курс се актуализира с `MAX_IN_COURSE`, в `courses` се добавят копия на `course`, като общият брой на обучавани в актуализирания и новосъздадените курсове да е първоначално откритата стойност за `numOfStudents`. Текстовото представяне на всяко от тези копия да се запише **thread-safe** в текстовия файл.

8 точки

7. Създайте нов WPF (.NET) проект (към същия *Solution*), *EducationClient* с `Output type` `Windows Application` за приемане на заявки на курсове със съответния брой обучавани.

Създайте WPF графичен интерфейс, който да възпроизвежда следния модел,



където:

- `Label Course Title` и `ComboBox cboTitles` служат за избиране на код от падащ списък с наименования на курсове
- `Label Course Type` и `ComboBox cboTypes` служат за избиране на код от падащ списък с типове на курсове
- `Label Number of students` и текстово поле `txtQty` служат за въвеждане на брой обучавани (по подразбиране дефинирайте 0)
- `Button Order` и `Button Cancel`

При изграждане на потребителския интерфейс да се спазят следните

**изисквания:**

- Да се ползва коректно вложение на WPF контроли от тип `Panel` и тип `Content`
- Да се ползват `Style` декларации за форматиране на `Label`, `ComboBox` и `Button`
- Да се въведат смислени идентификатори за рефериране на WPF контроли в съответствие с Модифицираната Унгарска нотация.

30 точки

8. Добавете метод към C# класа, съпътстващ XAML файла на графичния прозорец

```
void RunClient()
```

Нека този метод да създава връзка на `TcpClient` със сървър на IP `127.0.0.1` зададения номер на порт.

След установяване на връзка:

- да се създадат инстанции на `NetworkStream`, `BinaryReader` и `BinaryWriter`.
- да се прочетат `string` обектите, изпратени от сървъра с елементите на списъка със заглавия на курсове `titles` и константите на `ServiceType`, съдържащи тези данни, отделени помежду им със запетай
- да се **инициализират thread-safe** елементите на `cboTitles` и `cboTypes` със съответните данни, представени в прочетените два `string` обекта

Добавете прихващане на изключения и затваряне на ресурсите, които съответно възникват и се използват за комуникация с клиентите.

10 точки

9. Дефинирайте обработка на събитието `Click` за бутона `Order` на WPF прозореца, при което на сървъра се изпраща `string` с данните, въведени от клиента, всяка от тях разделена от останалите със запетая. Подредбата на данните да е следната:

- а. **Заглавието** (`Title`) на WPF прозореца,
- б. текущо избрания **елемент** в `cboTitles`
- в. текущо избрания **индекс** на елемент в `cboTypes`
- г. текущо **въведения брой студенти** в текстовия прозорец `txtQty`

4 точки

10. В конструктора на WPF прозореца, задайте стойност на заглавието (`Title`), което да се образува от текста „`Order client` ” и случайно генерирано число в интервала `[1, 100]`.

Създайте и стартирайте нишка за изпълнение на метода `RunClient()`

4 точки

