

Софийски Университет "Климент Охридски"

Факултет по Математика и Информатика

Финален изпит No. 1D

Курс: Обектно Ориентирано Програмиране с C#.NET

Преподавател: д-р. Е. Кръстев

Студент :

Дата:

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*

за решаване на следната задача.

Обработката на транзакции много често налага **търсене** на данни по определен признак (*lookup*). За тази цел е удобно да се създадат **библиотеки** с най- често използваните компоненти за такива приложения, при което отчетливо се **разграничават потребителския интерфейс, бизнес логиката и конкретното приложение**. В съответствие с тези цели създайте **client/ server** приложение за обработка на **файлове** като изпълните **следната последователност от действия на C#.NET**:

1. Създайте *UserControl (DLL)* с име *DataForma*, който **има две двойки (етикет, текстово поле)** и **два бутона (едно от текстовите полета да бъде RichText)**. Текстът, изобразяван от етикетите и заглавието на бутоните са данни на *DataForma* и се задават посредством **свойства (properties- get/ set)** за тези данни с подходящи **модификатори за достъп (access modifiers)**, така че всяка промяна в тези свойства **от потребителя на UserControl- а** да се отразява веднага на екрана.

Точки:16

2. Дефинирайте *delegate* и клас с аргументи на потребителско събитие *Save*. Дефинирайте това събитие в *DataForma* и го създайте (fire) при натискане на един от бутоните в *DataForma* (за определеност, **левият**), за да предостави за обработка на потребителя на този *UserControl* постоянния низ "save" и данните, въведени в **двете** текстови полета на *DataForma* като отделни полета от запис. Имената и структурата на делегата и класа с аргументите да са в съответствие с общоприетия стил за програмирането им.

Точки:11

3. Дефинирайте *delegate* и клас с аргументи на потребителско събитие *Request*. Дефинирайте това събитие в *DataForma* и го създайте (fire) при натискане на един от бутоните в *DataForma* (за определеност, **десният**), за да предостави за обработка на потребителя на този *UserControl* постоянния низ "request" и низа, въведен в **обикновеното** текстово поле на *DataForma*. Имената и структурата на делегата и класа с аргументите да са в съответствие с общоприетия стил за програмирането им.

Точки:11

4. Създайте **Windows** приложение на **многонишков file сървер**, който има текстово поле, където се описва диалога на операциите с клиент приложенията (**установяване, затваряне** на връзка с клиент и **типа** на текущо изпълняваната операция) . Операциите, извършвани от **сървера** се определят от **стойността на първия низ**, получен от **клиента**:

- a) Когато този низ е "request", тогава **сървера** проверява дали **втория низ**, изпратен от клиента е **име на съществуващ файл**. Когато този **файл съществува**, съдържанието на файла се **изпраща обратно на клиента**. Ако файл с такова **име не съществува**, на клиента се **изпраща съобщение**. **Връзката с клиента се затваря** след изпращане на файла или съобщението за несъществуващ файл.
- b) Когато този низ е "save", тогава **сървера** прочита **втория низ** от клиента като **име на файл**, в който да се **запише третия низ** изпратен от клиента. След записване на файла се **изпраща съобщение** на клиента за успешно/ неуспешното записване на файла. **Връзката с клиента се затваря** след изпращане съобщението до клиента.

Всяка връзка с клиент да се изпълнява от **отделна нишка**.

Точки:30

5. Създайте **Windows** клиент приложение, което има *DataForma UserControl* и допълнително **текстово поле** *txtLog* за отразяване на диалога със сървера. Задайте подходящи стойности за свойствата на *DataForma* (имена на етикети и бутони- "Filename", "File Contents", "Save", "Request" примерно)

a) Дефирайте метод *OnRequest* за обработка на събитието *Request*, което се създава с десния бутон на *DataForma*. Този метод създава *TCP stream socket* връзка със сървер приложението, описано по- горе. При установяване на връзка (*изписва се message в txtLog*), на сървера се изпращат двата низа, които са аргументите на събитието (постоянния низ "request" и низа, въведен в обикновеното текстово поле на *DataForma*, което е има смисъл на filename). Прочита се отговора на сървера и се изписва в *RichText* полето на *DataForma*, след което *връзката със сървера се затваря*. (*изписва се message в txtLog*)

b) Дефирайте метод *OnSave* за обработка на събитието *Save*, което се създава с левия бутон на *DataForma*. Този метод създава *TCP stream socket* връзка със сървер приложението, описано по- горе. При установяване на връзка (*изписва се message в txtLog*), на сървера се изпращат трите низа, които са аргументите на събитието (постоянния низ "save" и данните, въведени в двете текстови полета на *DataForma* като отделни полета от запис- които имат смисъл на filename и съдържание на file). Прочита се отговора на сървера и се изписва в *RichText* полето на *DataForma*, след което *връзката със сървера се затваря*. (*изписва се message в txtLog*)

Точки:22

6. Компилирайте и изпълнете приложението.

Точки:10