

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 8, 2022

Време за работа: 120 min

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

При установено преписване се присъждат нула точки за цялото контролно.

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:

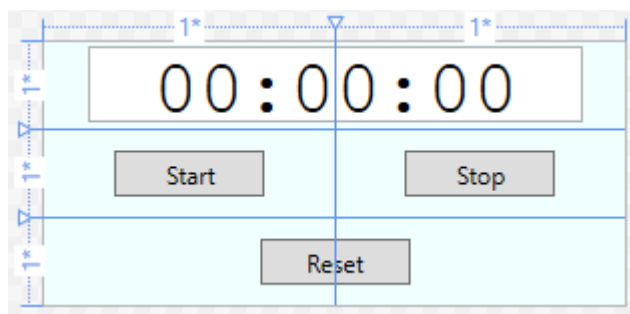
- **hiding of information**
- **software reuse**
- **inheritance**
- **polymorphism**

A. Създайте **WPF проект** *ClockLib* от тип **Class Library (.NET Framework)** на Visual Studio

1. **Напишете** class *ClockTickArgs*, който ще се използва за обект на събитие от тип *System.EventHandler*. Нека този клас има **C# get и set свойство** *ClockTick* от тип *System.ValueTuple*, представено като **наредена последователност данни** от вида **(int hour, int minute, int second)** и конструктор за общо ползване.

(5 точки)

2. **Добавете** към проекта *User control (WPF)* *DigitalClock*. Създайте в *DigitalClock* следния потребителски интерфейс



Да се използва подходяща йерархична структура от **WPF панели и контроли**, както и смислени наименования за идентификаторите на графичните компоненти с префикси по т. нар. **Модифицирана Унгарска нотация**.

(10 точки)

3. **Добавете** в съпътстващия class *DigitalClock*

- потребителски събития *ClockStarted* и *TimeUpdated* от тип *System.EventHandler* за абониране към съответни методи за обработка на тези събития.
- Данна *timer* от тип *DispatcherTimer*

(5 точки)

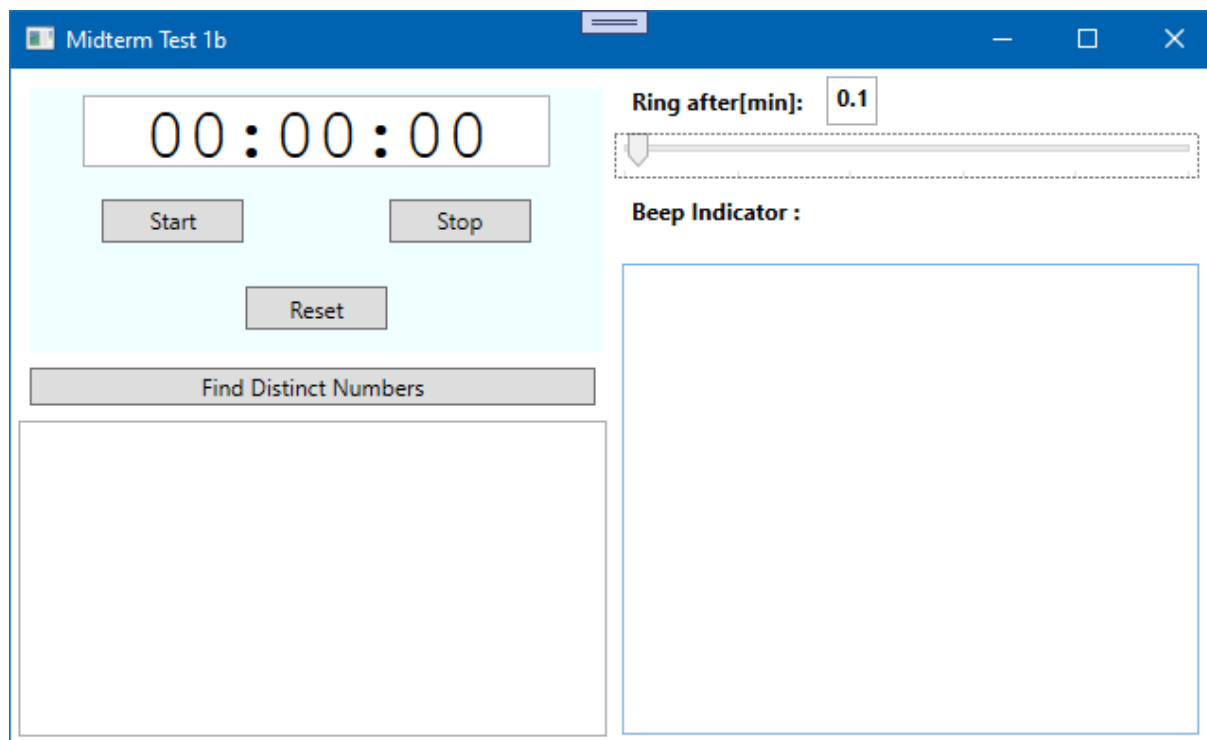
4. Реализирайте следната функционалност в графичната компонента *DigitalClock*:

- Да се инициализира *timer*, така че при стартирането му в текстовото поле на *DigitalClock* да се извежда текущото време (*DateTime.Now*) през 1 секунда и в показания формат. В края на всяка секунда да се обработва събитието *TimeUpdated* където обекта на събитието *ClockTickArgs* е инициализиран с текущите час, минута и секунда
- при натискане на бутона *Reset* да се изчиства текста в текстовото поле над групата от бутони
- при натискане на бутона *Start*
 - да се стартира *timer-a*
 - да се обработи събитието *ClockStarted* където обекта на събитието *ClockTickArgs* е инициализиран с текущите час, минута и секунда.
- при натискане на бутона *Stop* да се спира *timer-a*

(16 точки)

B. Добавете към същия Solution **нов проект**, наречен *AlarmClockApp* от тип **WPF App** ((.NET Framework)(64 т)

1. Създайте графичен прозорец в *AlarmClockApp* със следния дизайн:



където в горния ляв ъгъл да се вмъкне инстанция на потребителския контрол *DigitalClock*, разработен в т. А.. Да се използва подходяща йерархична структура от WPF панели и контроли, както и смислени наименования за идентификаторите на графичните компоненти с префикси по т. нар. Модифицирана Унгарска нотация.

(22 точки)

2. Добавете **Binding** (XAML или C# код) между свойствата *Value* и *Text* съответно на *Slider*-а (Binding source) и *TextBox*-а (Binding target) в горната дясна част на графичния прозорец. **Стойностите в текстовото поле да са форматирани** с един знак след десетичната запетая.

(8 точки)

3. Добавете следните данни към класа, съпътстващ графичния прозорец:
- Данни *startTime* и *currentTime* от тип *System.ValueTuple*, представени като **наредена последователност данни** от вида (*int hour*, *int minute*, *int second*)
 - Данна *rand* от тип *Random*
 - Данна *ringAfter* от тип *double*

(6 точки)

4. **Регистрирайте методи за обработка** на събитията *ClockStarted* и *TimeUpdated* на инстанцията на контролата *DigitalClock* (ползвайте C# или XAML код за абониране към събитието).
- Методът за обработка на *ClockStarted* **да инициализира данната *startTime*** като използва съответните стойности на *ClockTick*, съдържащи се обекта на събитието. Също така, **да инициализира *ringAfter*** със стойността, съдържаща се в текстовото поле след етикета *Ring After [min]*. **Изписва времето за стартиране** на часовника в текстовото поле под бутона *Find distinct numbers*
 - Методът за обработка на *TimeUpdated* **да инициализира данната *currentTime*** като използва съответните стойности на *ClockTick*, съдържащи се обекта на събитието. Също така, **на всяка четна секунда изписва** произволно избрано число от интервала [10, 50] и празен символ в текстовото поле под етикета *Beep Indicator*. **Проверява дали са изтекли *ringAfter* минути на *currentTime*** спрямо *startTime*. **След изтичане на това време изписва *Start ringing*** в етикета *Beep Indicator* и изпълнява командата *SystemSounds.Beep.Play()* за **извеждане на звуков сигнал**.

(16 точки)

5. **При натискане на бутона *Find distinct numbers*:**
- Да се изведат различните числа, генерирани след стартиране на часовника, където числата да са разделени с празен символ.. Списъкът с тези числа да се прочете от многоредовото текстово поле, намиращо се отдясно на бутона. текстовото поле За решението на тази задача **да се използва LINQ**. (виж примера)

- Да се изведе също статистика колко пъти се среща всяко число измежду числата в многоредовото текстово поле, намиращо се отдясно на бутона. Числата да са сортирани в низходящ ред. За решението на тази задача **да се използва LINQ**. (виж примера)

(12 точки)

Midterm Test 1b

14:31:40

Start Stop

Reset

Find Distinct Numbers

Start time: 14:31:23

Ring after[min]: 0.5

14 37 29 25 43 13

Midterm Test 1b

14:32:59

Start Stop

Reset

Find Distinct Numbers

Start time: 14:31:23

Ring after[min]: 0.5

Start ringing.

14 37 29 25 43 13 17 17 24 32 34 11 30 41 21 26 19 41 21

Distinct numbers
14 37 29 25 43 13 17 24 32 34 11 30 41 21 26 19

Frequency
Number:43 found 1 times.
Number:41 found 2 times.
Number:37 found 1 times.
Number:34 found 1 times.
Number:32 found 1 times.
Number:30 found 1 times.
Number:29 found 1 times.