|  |
| --- |
| , RD Dep. |
| Практическое задание  NET.C#.10 Делегаты и события |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| REVISION HISTORY | | | | | |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Effective Date |
| 1.0 | Initial version | Анжелика Кравчук |  |  |  |
| 1.1 | Review and corrections. | Владимир Тихон |  |  |  |
| 1.2 | Added details | Oleg |  |  |  |

**Задание.**

* Разработать класс для имитации часов с обратным отсчетом (таймер), реализующий возможность по истечении назначенного времени (количество секунд ожидания, поставляются пользователем) передавать сообщение любому подписавшемуся на событие типу.
* Для создания эффекта ожидания использовать метод Thread.Sleep пространства имен System.Threading.
* Пользователь может задавать имя таймера.
* Доработать тип делегата, чтобы получатель события мог определить «источник события» и вывести «имя таймера».
* Добавить событие с отсчётом времени, в котором будет передаваться «сколько секунд осталось».
* Предусмотреть возможность подписки на событие в нескольких классах.
* Создайте три пользовательских класса явно реализующие интерфейс ICutDownNotifier с методами Init() (подписывается на событие «таймера») и Run(запускает «таймер»):
  + Один обрабатывает события с помощью методов;
* Второй обрабатывает события с помощью анонимных делегатов;
* Третий обрабатывает события с помощью лямбда выражений.
* В итого каждый из пользовательских класссов должен обрабатывать три события:
  + Старт обратного отсчёта;
  + Осталось N-секунд;
  + Обратный отсчёт завершён.
* Каждый обработчик должен выводить «имя таймера» в обработчиках события.
* Создайте делегат «Началось время выполнения задания» с параметрами «Название задачи», «Сколько было отведено времени».
* Конструктов каждого из пользовательских классов, реализующих ICutDownNotifier, должен принимать два параметра:
  + Созданный делегат «Началось время выполнения задания»;
  + Делегат «Закончилось время выполнения задания» на основе стандартного типа Action<T> с параметрами «Название задачи», «Сколько было отведено времени».
* Эти делегаты нужно вызывать в соответствующее время.
* Создайте в приложении методы, который будет информировать о начале и завершении времени выполнения заданий, и передавайте их в конструкторы «пользовательских классов».
* Приложение должно одновременно настроить и запустить все три таймера с названиями ««Чтение задание», «Выполнение задания» и «Проверка задания перед отправкой».
  + При этом надо работать с массивом объектов типа ICutDownNotifier.
* В качестве UI-интерфейса использовать консольное приложение с интерфейсом командной строки, WinForms или WPF-приложение.

**Важно:**

* Применить правила оформления событий

<http://www.rsdn.ru/article/mag/200401/codestyle.xml#EVOAC>

* How to: Publish Events that Conform to .NET Framework Guidelines (C# Programming Guide)

<http://msdn.microsoft.com/en-us/library/w369ty8x.aspx>

* Framework Design Guidelines: Events and Callbacks

<http://msdn.microsoft.com/en-us/library/ms229041.aspx>

* Action<T> Delegate

<http://msdn.microsoft.com/ru-ru/library/018hxwa8.aspx>