|  |  |
| --- | --- |
|  | Делегаты и события. Задания |
| EPMR-TCSR |

ИСТОРИЯ ИЗМЕНЕНИЙ

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Версия | Описание изменений | Автор | Дата | Утверждено   |  |  | | --- | --- | | Имя | Дата | | |
| <1.0> | Initial version | Dmitry\_Vereskun | <05-Mar-2012> |  |  |
|  |  |  |  |  |  |

СВЯЗАННЫЕ ДОКУМЕНТЫ

|  |  |
| --- | --- |
| Номер | Название документа |
|  |  |
|  |  |

# Задания

Выполните приведенные ниже задания. Задания, помеченные «\*», являются заданиями повышенной сложности и не обязательны для выполнения.

# Задание 1

Написать программу, выполняющую сортировку массива строк по возрастанию длины. Если строки состоят из равного числа символов, их следует отсортировать по алфавиту. Реализовать метод сравнения строк отдельным методом, передаваемым в сортировку через делегат.

# Задание 2

Написать программу, описывающую небольшой офис, в котором работают сотрудники – объекты класса Person, обладающие полем имя (Name). Каждый из сотрудников содержит пару методов:

1. приветствие сотрудника, пришедшего на работу (принимает в качестве аргументов объект сотрудника и время его прихода). В зависимости от времени суток, приветствие может быть различным: до 12 часов – «Доброе утро», с 12 до 17 – «Добрый день», начиная с 17 часов – «Добрый вечер».
2. прощание с ним (принимает только объект сотрудника).

Каждый раз при входе очередного сотрудника в офис, все пришедшие ранее его приветствуют. При уходе сотрудника домой с ним также прощаются все присутствующие. Вызов процедуры приветствия/прощания производить через групповые делегаты. Факт прихода и ухода сотрудника отслеживается через генерируемые им события.

Событие прихода описывается делегатом, передающим в числе параметров наследника EventArgs, явно содержащего поле с временем прихода.

Продемонстрировать работу офиса при последовательном приходе и уходе сотрудников.

Пример:

# 

# Задание 3\*

Написать модуль сортировки, включающий в себя:

* Метод сортировки из задания 1;
* Метод, позволяющий запустить в сортировку в отдельном потоке выполнения;
* Событие, сигнализирующее о завершении сортировки.

Продемонстрировать работу модуля в многопоточном режиме.

http://www.albahari.com/threading/

# Требования к оформлению

1. Для работы автоматических фильтров необходимо, чтобы тема письма выглядела следующим образом: [STYYYY\_N]Фамилия\_Имя\_TaskNN. Здесь ST – сокращенное наименование программы (Students Training), YYYY – номер года, N – номер группы, NN – порядковый номер задания в виде двух цифр, напр. 01, 05, 12 и т.д.
2. Каждое задание должно представлять собой отдельный project в рамках общего solution. Название проектов: Task1, Task2 и т.д. Название solution должно формироваться по принципу: Фамилия\_Имя\_Task1.
3. Проект должен быть прислан в zip архиве (rar, 7z и др. не допускаются). Имя архива должно совпадать с именем solution.
4. Не ленитесь сделать ввод/вывод дружественным по отношению к пользователю. Программы, выводящие мигающий курсор на черном фоне, сразу получают «ноль».