EPAM Systems, RD Dep.

Homework Task

08 – Extended C#

REVISION HISTORY					
Ver.	**Description of Change**	**Author**	**Date**	**Approved**	
				Name	**Effective Date**
1.0	Initial version	Dmitry_Vereskun	05.03.2012		
2.0		Nikolay Piskarev	29.05.2017		

Задание 1

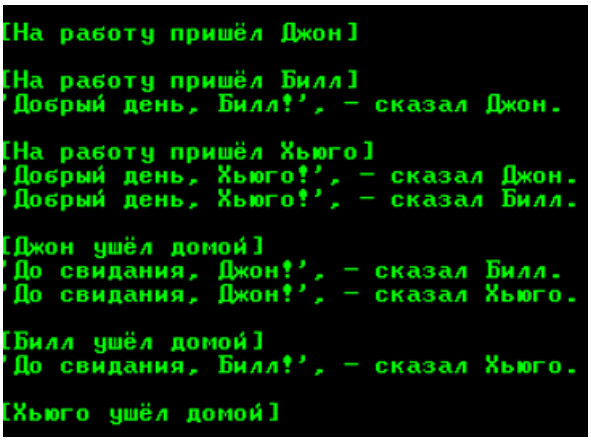
Написать программу, выполняющую сортировку массива строк по возрастанию длины. Если строки состоят из равного числа символов, их следует отсортировать по алфавиту. Реализовать метод сравнения строк отдельным методом, передаваемым в сортировку через делегат.

Задание 2

Написать программу, описывающую небольшой офис, в котором работают сотрудники – объекты класса Person, обладающие полем имя (Name). Каждый из сотрудников содержит пару методов: приветствие сотрудника, пришедшего на работу (принимает в качестве аргументов объект сотрудника и время его прихода) и прощание с ним (принимает только объект сотрудника). 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». 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:



Требования к оформлению

1. Для каждой домашней работы необходимо создать собственную папку в корне репозитория.
2. Каждое задание должно представлять собой отдельный проект (project) в рамках общего решения (solution).
3. Правила именования:
   Все названия и имена должны быть переведены на английский язык. Именование транслитерацией не допускается.
   a. Маска именования репозитория: EPAM_Ext_Lab_<Квартал старта группы>_<Год>_<Фамилия_Имя>.
   Например, EPAM_Ext_Lab_Q2_2017_Ivanov_Ivan
   b. Маска именования решения: HMT_<Порядковый номер>. *Например, HMT_01, HWT_11.*
   c. Маска именования проектов: <Задача><Номер задачи>. *Например, Task01, Task12*.
   d. Каждое задание должно быть оформлено следующим образом:
   1. Формулировка задания в виде комментария (просто копировать-вставить из этого файла).
   2. Код, решающий задачу.
4. Написать письмо преподавателю, информирующее о том, что домашняя работа выполнена. Тема письма должна выглядеть так: [STYYYY_N]Фамилия_Имя_TaskNN. Здесь ST – сокращённое 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.

   Например, ST2017_1 Иванов Иван Task01.

Рекомендации к оформлению кода

1. При добавлении нового файла желательно вверху файла добавить блок с комментарием, для чего предназначен этот файл.
2. При добавлении метода в блоке <summary></summary> описать предназначение метода, его входных/выходных параметров и возвращаемых значений.
3. Использовать утилиту StyleCop для проверки правил оформления кода. Перед отправкой задания на проверку запускать StyleCop и исправлять все предупреждения, возникающие в секции Warning Visual Studio. Предупреждения об отсутствии документации можно игнорировать (либо настроить StyleCop таким образом, чтобы он не проверял документирование кода).