

Инструменты и средства программирования

Лабораторная работа №8

Файлы и потоки данных

Цель работы: знакомство с классами пространства имен System.IO.

Задача работы: Научиться выполнять основные операции над объектами файловой системы. Научиться выполнять чтение/запись из/в файл.

Время выполнения работы: 2 часа

Результат выполнения работы: программа, обеспечивающая функционал согласно заданию.

1. Задание

- a) Создать новый проект
- b) Описать класс *Employee* (сотрудник), содержащий любые свойства типа *int* и *bool* и свойство *Name* (имя) типа *string*
- c) Описать класс *FileService*, реализующий интерфейс:

```
interface IFileService
{
    IEnumerable<Employee> ReadFile(string fileName);
    void SaveData(IEnumerable<Employee> data, string fileName);
}
```

Метод `ReadFile` – именованный итератор (оператор *yield return*), считывающий данные из бинарного файла с именем `fileName`

Метод `SaveData` сохраняет коллекцию `data` в бинарный файл с именем `filename`

Запись и чтение файла выполнить с помощью `BinaryReader/BinaryWriter`

Файловый поток после чтения/записи должен быть уничтожен (использовать оператор *using*)

d) Описать класс `EmployeeComparer<Employee>:IComparer<Employee>`, позволяющий сравнивать объекты класса `Employee` по свойству `Name`

e) В классе `Program`:

- Создать коллекцию объектов класса `Employee`. Заполнить коллекцию 5-6 объектами
- Создать файл для записи с произвольным именем. Если такой файл уже существует, предварительно удалить существующий файл.
- С помощью класса `FileService` записать в созданный файл коллекцию объектов класса `Employee`.
- Переименовать файл.
- Создать пустую коллекцию объектов класса `Employee`. С помощью класса `FileService` заполнить коллекцию данными из файла с новым именем.
- Отсортировать полученную коллекцию с помощью запроса LINQ (использовать класс `EmployeeComparer`)
- Вывести в консоль содержимое исходной коллекции и отсортированной коллекции, прочитанной из файла