# Практическое занятие № 20. Использование интерфейсов

**1 Цель занятия**

Получить практические навыки реализации интерфейсов

**2 Перечень оборудования и программного обеспечения**

Персональный компьютер

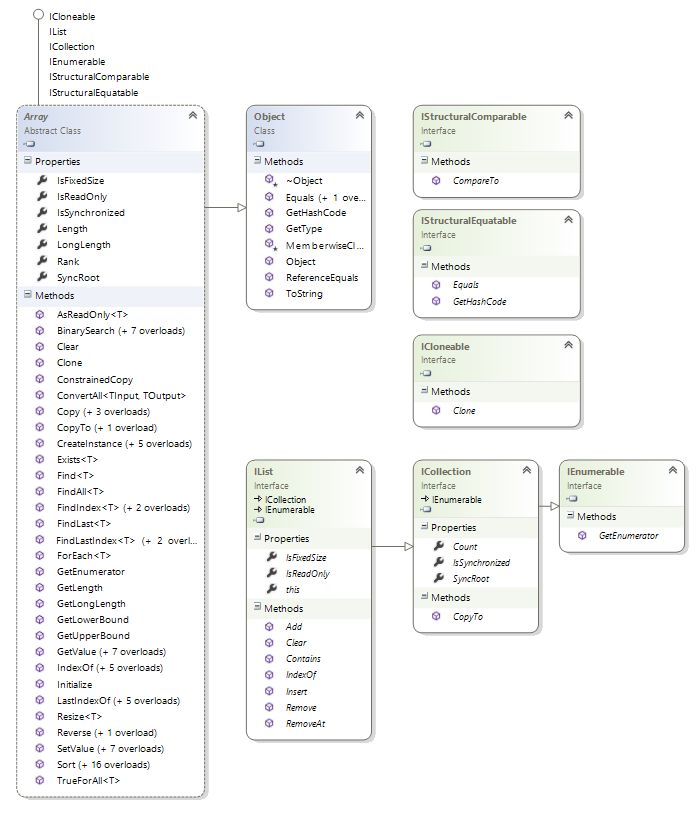
Microsoft Office (Word)

Microsoft Visual Studio

**3 Краткие теоретические сведения**

***Интерфейс***представляет собой полностью абстрактный (неконкретный) класс, все методы которого абстрактны. Методы интерфейса объявляются без указания модификатора доступа, и класс, наследующий интерфейс, обязан полностью реализовать все методы интерфейса. Этим класс, наследующий интерфейс, отличается от класса, наследующего абстрактный класс, где потомок может реализовать лишь некоторые методы родительского абстрактного класса.

Так, все классы, являющиеся массивами, являются потомками абстрактного класса Array:



Класс Array наследует ряд интерфейсов: ICloneable, IList, ICollection, IEnumerable, и следовательно, обязан реализовать все их методы и свойства. Помимо наследования свойств и методов класса Object и вышеперечисленных интерфейсов, класс Array имеет довольно большое число собственных методов и свойств.

Интерфейс позволяет описывать некоторые желательные свойства, которыми могут обладать объекты разных классов.

Среди интерфейсов, встроенных в библиотеку базовых классов .NET, можно особо выделить такие как:

* IEnumerable (для работы с наборами объектов, в т. ч. с использованием оператора foreach);
* IClonable (копирование объектов);
* IComparable (для сравнения и сортировки объектов).

Пространство имен System.Collections, предназначенное для работы с наборами объектов, поддерживает интерфейсы:

* ICollection (определяет общие характеристики класса набора элементов);
* IComparer, IDictionary (позволяет представлять содержимое объекта в виде пар имя-значение);
* IDictionaryEnumerator (нумерация содержимого объекта, поддерживающего IDictionary);
* IEnumerable, IEnumerator (позволяет циклически обращаться к элементам набора объектов);
* IHashCodeProvider (возвращает хэш-код с помощью выбранного алгоритма хеширования);
* IList (обеспечивает методы добавления, удаления и индексирования элементов в списке объектов).

Для циклического обращения к элементам коллекции зачастую проще (да и лучше) организовать цикл foreach, чем пользоваться непосредственно методами интерфейса IEnumerator. Тем не менее, ясное представление о принципе действия подобных интерфейсов важно иметь по следующей причине: если требуется создать класс, содержащий объекты, перечисляемые в цикле foreach, то в этом классе следует реализовать интерфейсы IEnumerator и IEnumerable. Иными словами, для того чтобы обратиться к объекту определяемого пользователем класса в цикле foreach, необходимо реализовать интерфейсы IEnumerator и IEnumerable в их обобщенной или необобщенной форме.

Большинство коллекций реализует интерфейс IEnumerable<Тип> явно или неявно. Тип Array наследует интерфейс IEnumerable.

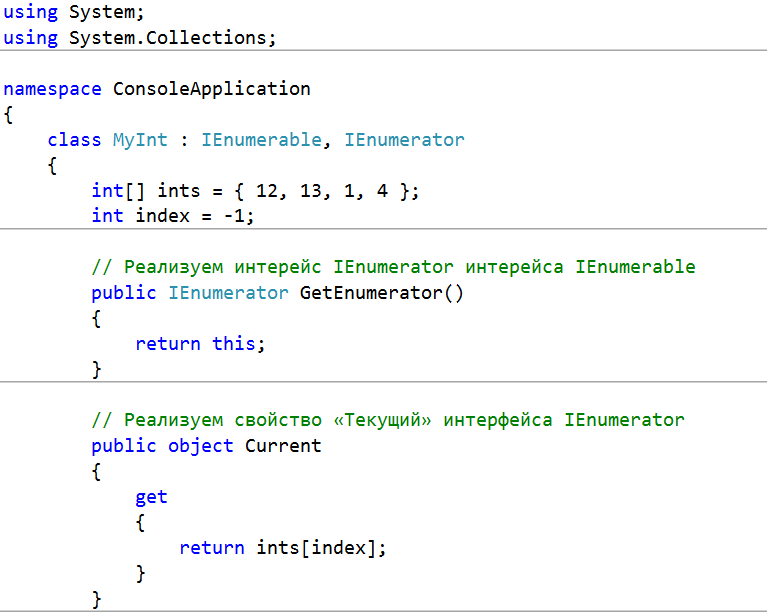
IEnumerable подходит для перебора по коллекции. Вы не можете изменить (добавить или удалить) данные из IEnumerable.

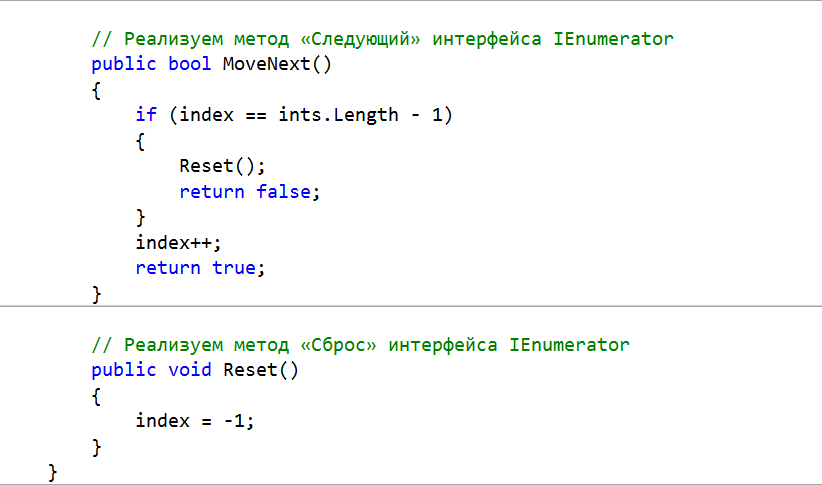
IEnumerable может двигаться **только вперед** по коллекции, он не может идти назад.

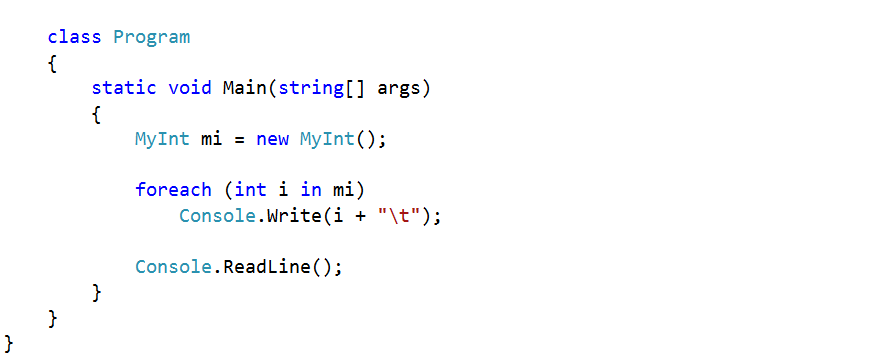
Для использования интерфейсов IEnumerator и IEnumerable необходимо подключение System.Collections.

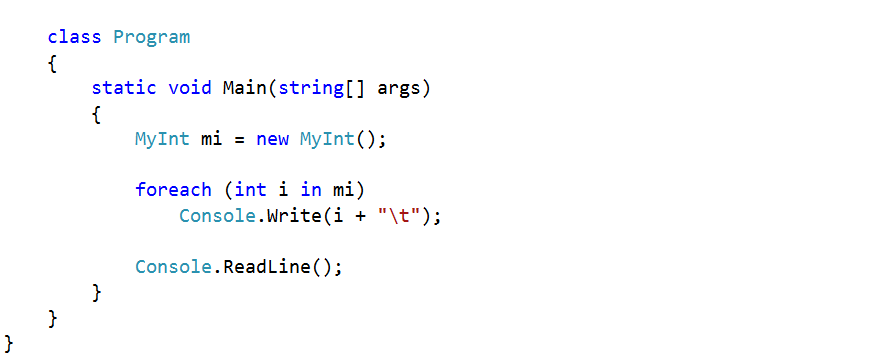
В рассматриваемом далее примере создается класс MyInt, содержащий некоторый массив и наследующий интерфейсы IEnumerable, IEnumerator.

**Пример 1.** Организация циклического обращения к элементам класса MyInt, содержащего массив и наследующего интерфейсы IEnumerable, IEnumerator:











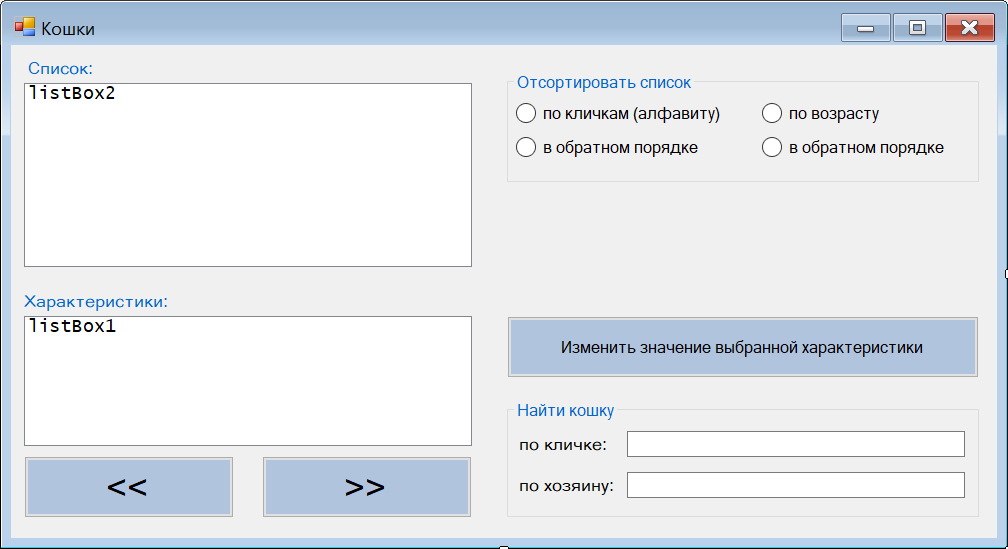
В рассматриваемом далее задании показано использование интерфейсов IComparable и IComparer для сравнения и сортировки объектов класса Cat. Эти интерфейсы имеет всего по одному методу: интерфейс IComparable<Cat> имеет метод CompareTo(Cat obj), предна­значенный для сравнения текущего объекта класса Cat с другим объектом этого же класса; интерфейс IComparer<T> where T : Cat имеет метод Compare(T x, T y), предназначенный для сравнения двух объектов, насле­дующих класс Cat. На выходе методы CompareTo() и Compare() возвращают целое число, которое может иметь одно из трех значений:

* меньше нуля (значит, текущий объект должен находиться перед объектом, который передается в качестве параметра);
* равен нулю (значит, оба объекта равны);
* больше нуля (значит, текущий объект должен находиться после объекта, передаваемого в качестве параметра).

**Задание.** Реализовать интерфейсы в классах Cats и Cat, разработанных в рамках практического занятия № 22. С помощью реализации интерфейсов IEnumerable и IEnumerator организовать циклическое обращение к объектам класса Cats для вывода их данных в список. С помощью реализации интерфейсов IComparable и IComparer организовать сортировку объектов класса Cats по двум критериям.

**Решение.**

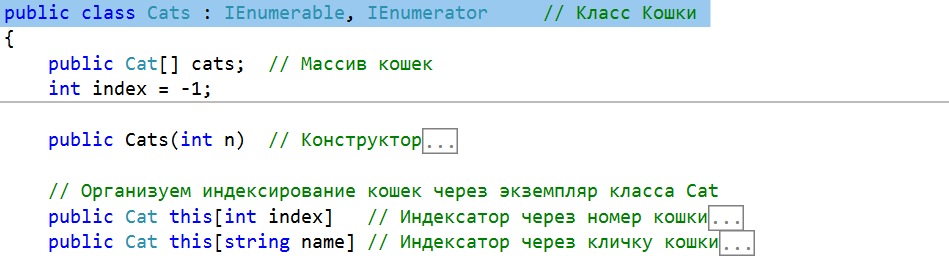
Форма Windows-приложения для представления и сортировки объектов заданных классов:

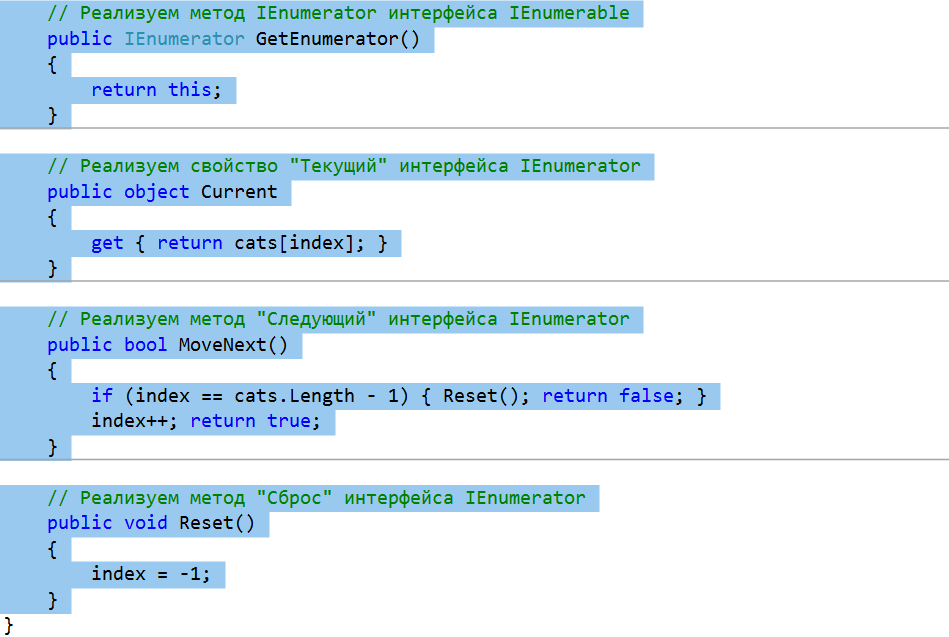


Класс Cats, с реализаций интерфейсов IEnumerable и IEnumerator для организации циклического обращения к объектам класса:



…

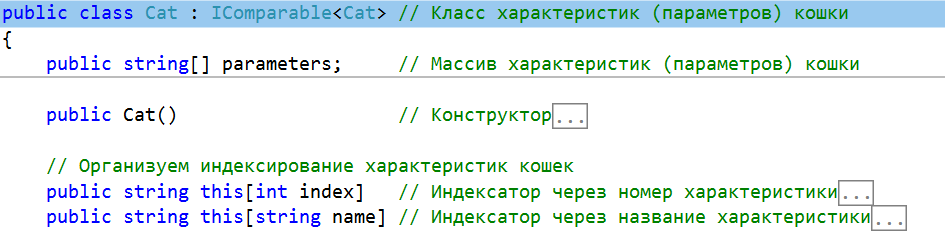


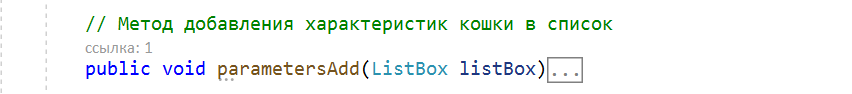


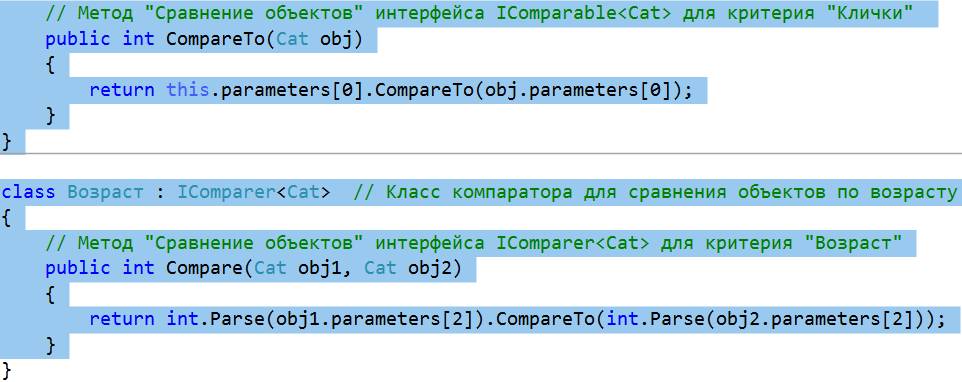
Класс Cat с реализаций интерфейсов IComparable и IComparer для организации сортировки его объектов по критериям «Клички» и «Возраст»:

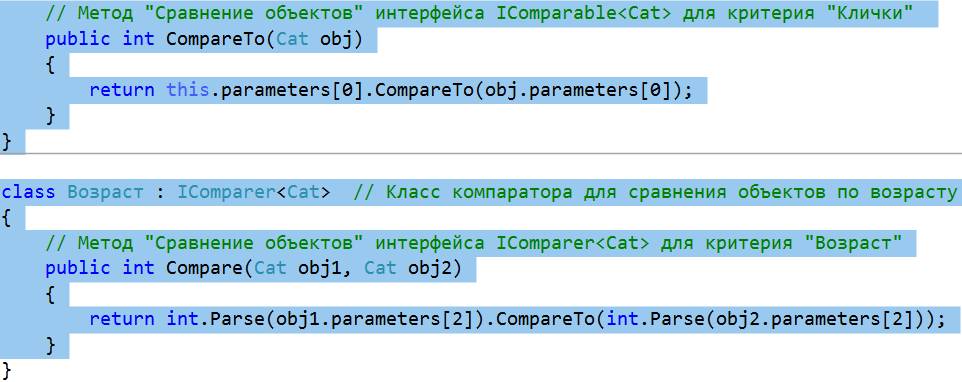


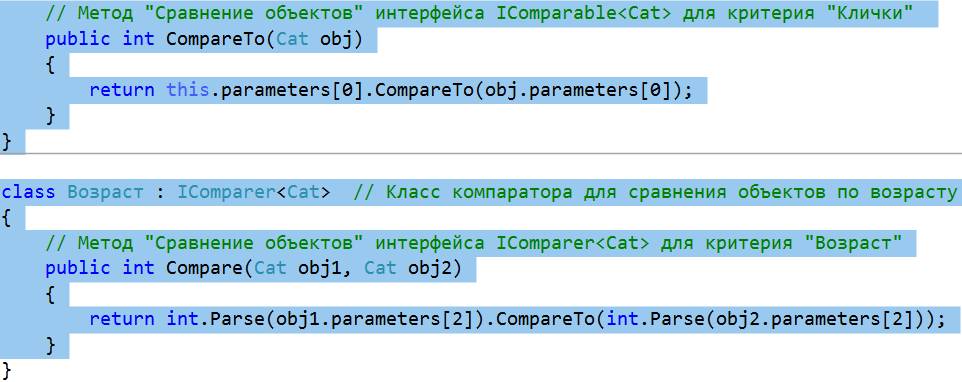
…



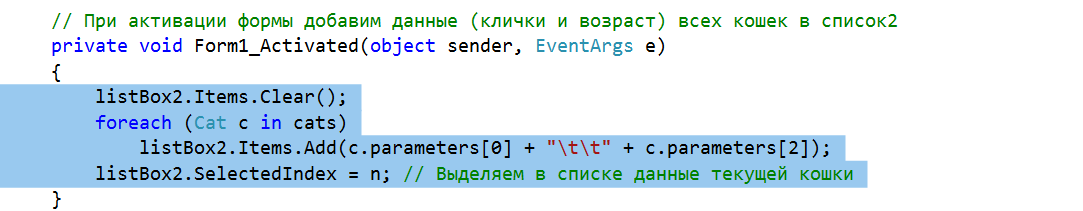


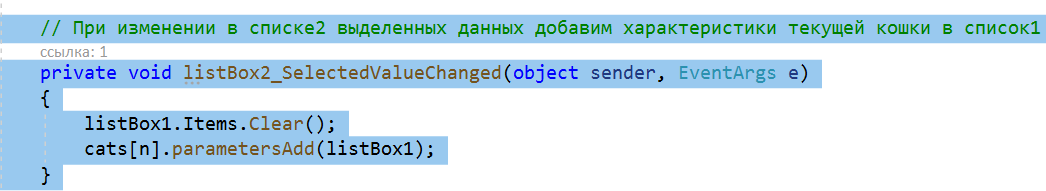


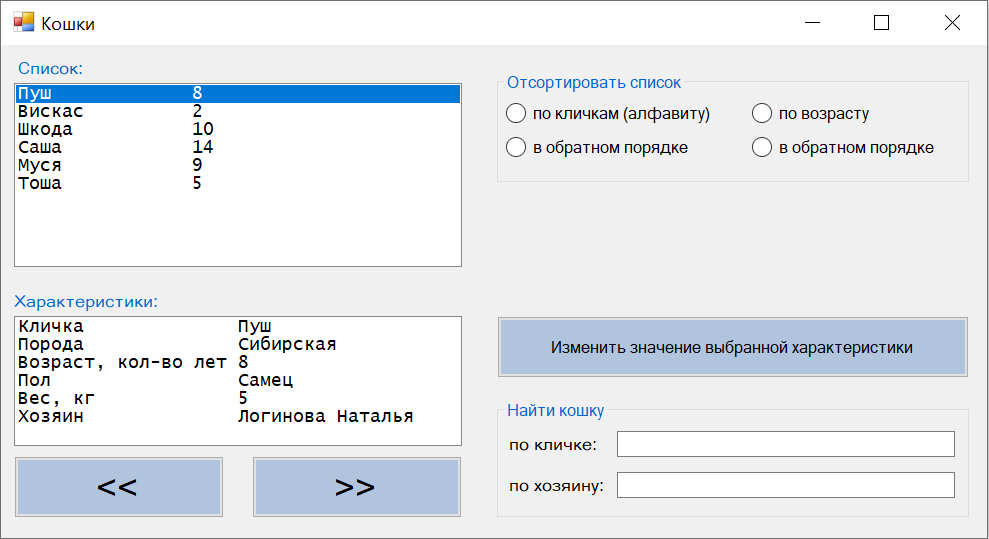


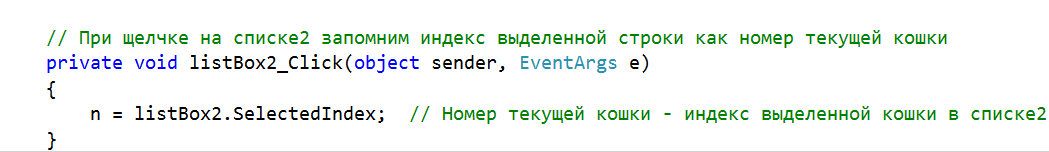


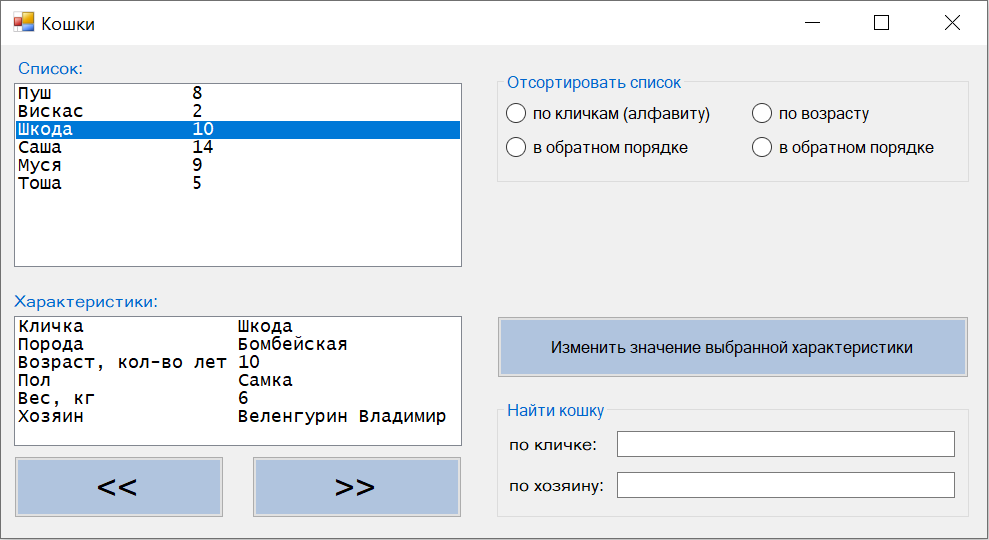
Коды методов класса Form1, заносящих данные (характеристики) созданных классов в списки:



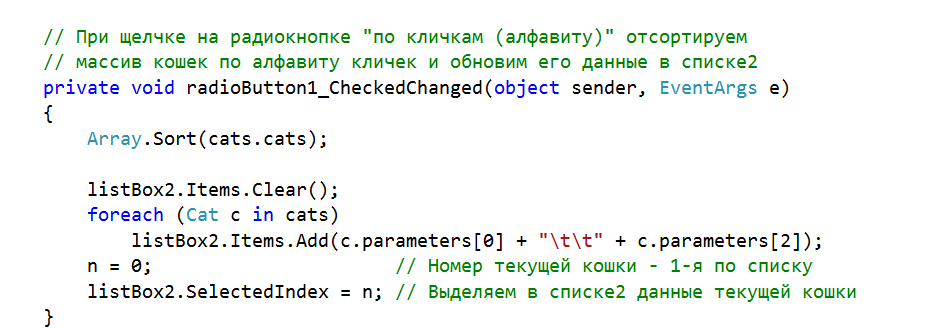


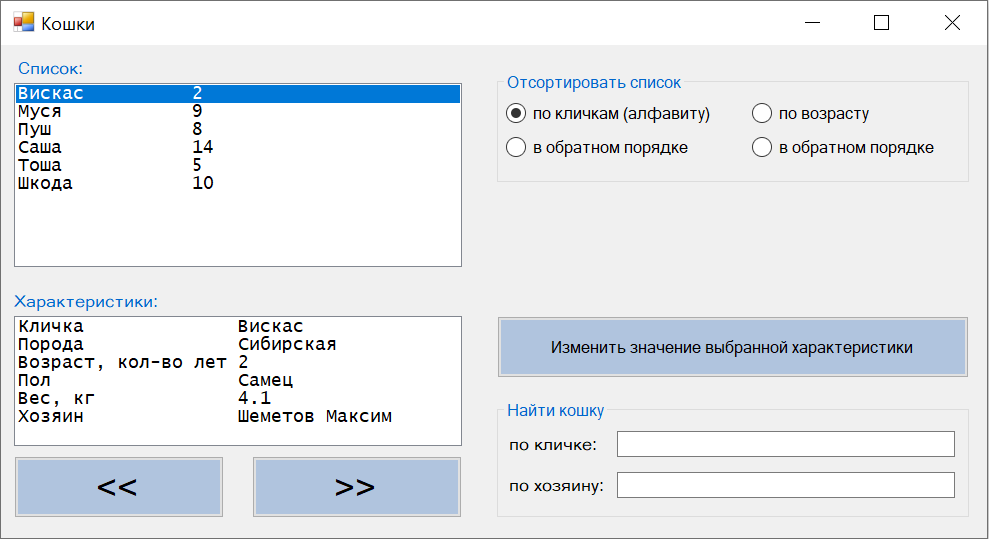


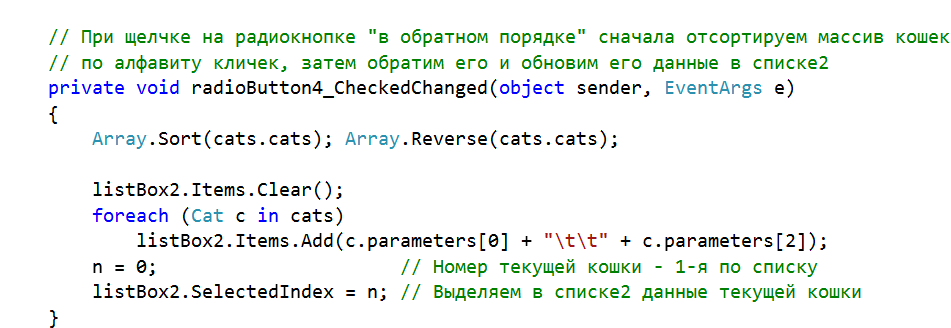


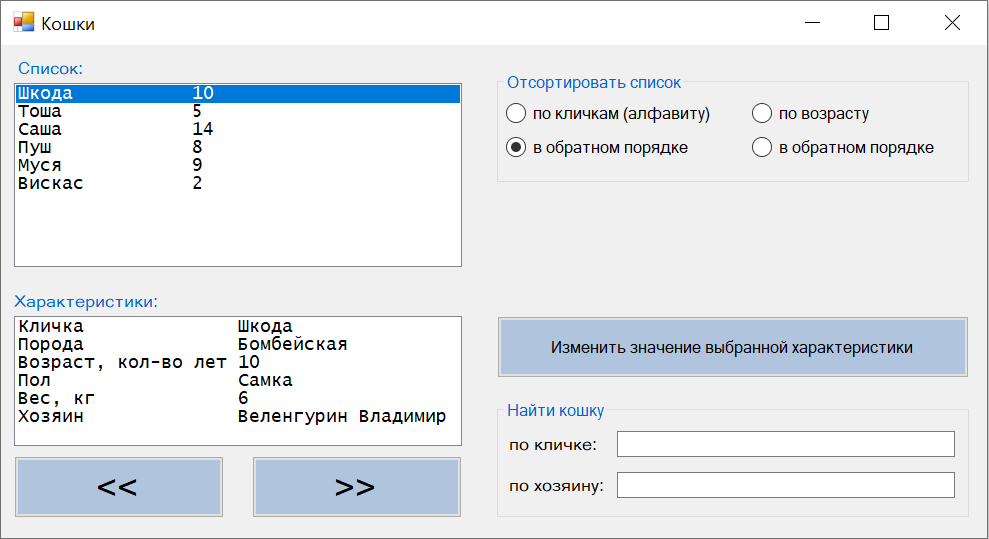


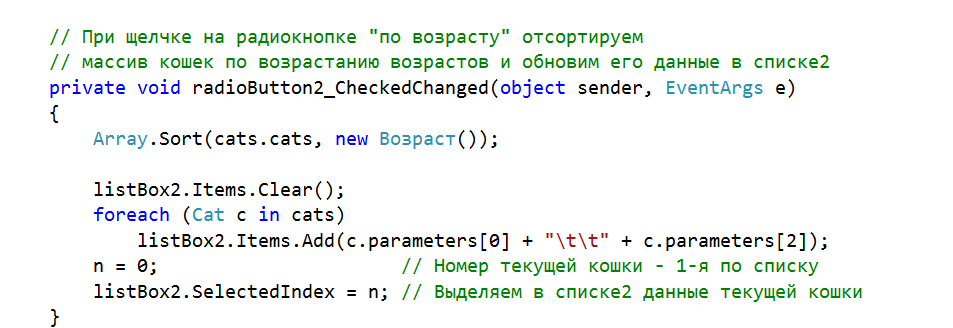
Методы класса Form1 для сортировки объектов класса Cats по двум критериям:

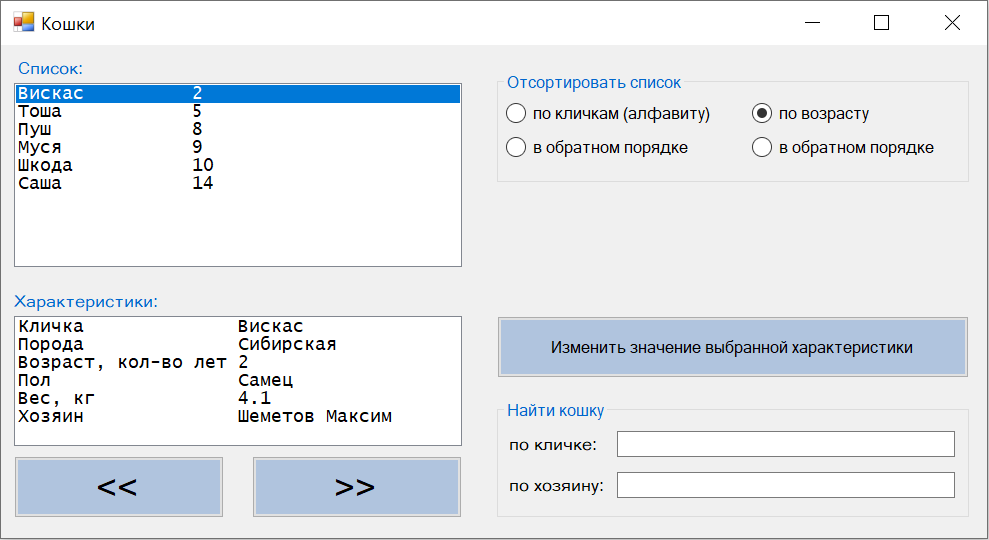


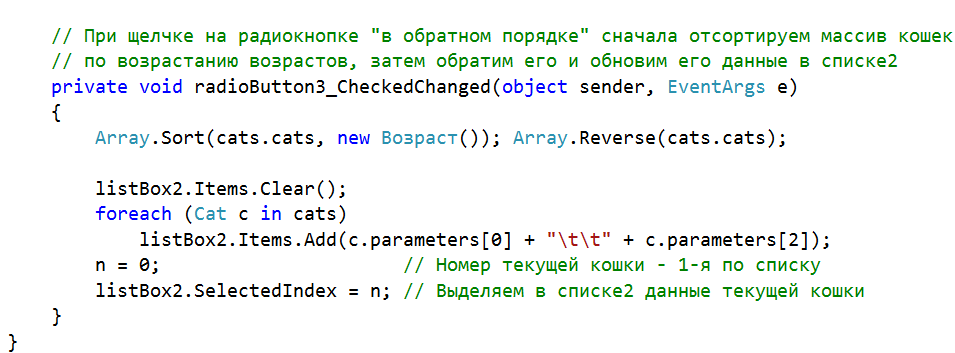


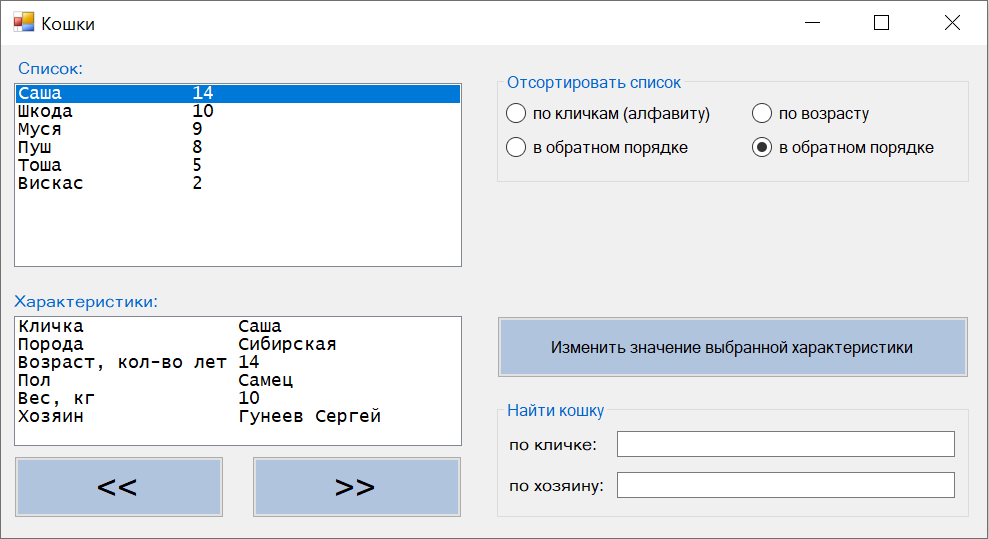












**4 Порядок выполнения работы**

4.1 Изучить теоретические сведения и задание к работе.

4.2 В соответствии с вариантом задания реализовать в классах, разработанных в рамках практического занятия № 19, интерфейсы IEnumerable и IEnumerator (для организации циклического обращения к объектам класса) и интерфейсы IComparable и IComparer (для организации сортировки объектов класса по двум критериям).

4.3 Доработать и отладить методы занесения данных (характеристик) объектов созданных классов в списки: Form\_Activated() – для отображения в списке данных всех объектов класса; listBox\_Click() – для запоминания индекса выделенной строки в этом списке как номера текущего объекта класса; listBox\_SelectedValueChanged() – для отображения в другом списке характеристик текущего объекта класса.

4.4. Разработать методы для сортировки массива объектов класса по двум критериям.

**5 Содержание отчета**

5.1 Название работы

5.2 Цель работы

5.3 Программные коды и скриншоты работы классов и методов разработанного приложения Windows Forms в соответствии с вариантом задания

**6 Варианты заданий**

6.1 Реализовать интерфейсы для циклического перебора и сортировки объектов классов STUDENTS и STUDENT, представляющих информацию об успеваемости студентов по разным дисциплинам за семестр.

6.2 Реализовать интерфейсы для циклического перебора и сортировки объектов классов EMPLOYEES и EMPLOYEE, представляющих информацию о заработной плате сотрудников за год.

6.3 Реализовать интерфейсы для циклического перебора и сортировки объектов классов ATHLETES и ATHLETE, представляющих информацию о физических данных спортсменов.

6.4 Реализовать интерфейсы для циклического перебора и сортировки объектов классов CARS и CAR, представляющих информацию о параметрах автомобилей.

6.5 Реализовать интерфейсы для циклического перебора и сортировки объектов классов BORROWERS и BORROWER, представляющих информацию по предоставляемым заемщикам кредитам.

6.6 Реализовать интерфейсы для циклического перебора и сортировки объектов классов PATIENTS и PATIENT, представляющих информацию о данных пациентов.

6.7 Реализовать интерфейсы для циклического перебора и сортировки объектов классов RATES и RATE, представляющих информацию о курсах разных валют за неделю.

6.8 Реализовать интерфейсы для циклического перебора и сортировки объектов классов PRODUCTS и PRODUCT, представляющих информацию о товарах в магазине.

6.9 Реализовать интерфейсы для циклического перебора и сортировки объектов классов BUYERS и BUYER, представляющих информацию о расходах покупателей за неделю.

6.10 Реализовать интерфейсы для циклического перебора и сортировки объектов классов COLLEGES и COLLEGE, представляющих информацию о звонках на перемену в колледжах.

6.11 Реализовать интерфейсы для циклического перебора и сортировки объектов классов BUILDINGS и BUILDING, представляющих информацию о параметрах продаваемых домов.

6.12 Реализовать интерфейсы для циклического перебора и сортировки объектов классов SESSIONS и SESSION, представляющих информацию о начале сеансов фильмов в кинотеатре.

6.13 Реализовать интерфейсы для циклического перебора и сортировки объектов классов EXAMS и EXAM, представляющих информацию о количестве оценок разного типа на экзаменах.

6.14 Реализовать интерфейсы для циклического перебора и сортировки объектов классов TEACHERS и TEACHER, представляющих информацию о данных преподавателей.

6.15 Реализовать интерфейсы для циклического перебора и сортировки объектов классов LIBRARIES и LIBRARY, представляющих информацию о количестве разных книг в библиотеках.

6.16 Реализовать интерфейсы для циклического перебора и сортировки объектов классов LOCALITIES и LOCALITY, представляющих информацию о погодных условиях в населенных пунктах.

6.17 Реализовать интерфейсы для циклического перебора и сортировки объектов классов INSTITUTIONS и INSTITUTION, представляющих информацию о телефонах образовательных учреждений.

**7 Используемая литература**

7.1 Г. Гагарина, Е. В. Кокорева, Б. Д. Виснадул Технология разработки программного обеспечения. Форум, Инфра-М, 2009

7.2 Эндрю Троелсен Язык программирования С# 2010 и платформа .NET 4.0, М., ВИЛЬЯМС, 2011

7.3 А. Марченко C#. Введение в программирование, М, Вильямс, 2009

7.4 http://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx.