# Практическое занятие № 21. Разработка модулей для работы с хеш-таблицами

**1 Цель занятия**

Получить практические навыки разработки модулей для работы с хеш-таблицами

**2 Перечень оборудования и программного обеспечения**

Персональный компьютер

Microsoft Office (Word), Microsoft Visual Studio

**3 Краткие теоретические сведения**

**3.1 Структура (коллекция) Hashtable и ее применение**

***Хеш-таблица*** — это структура данных, реализующая интерфейс ассоциативного массива, а именно, она позволяет хранить пары (ключ, значение) и выполнять три операции: операцию добавления новой пары, операцию поиска и операцию удаления пары по ключу.

Класс Hashtable предоставляет коллекцию пар "ключ-значение", которые упорядочены по хэш-коду ключа.

**3.2 Свойства хеш-таблицы**

Каждый элемент в паре "ключ-значение" хранится в объекте DictionaryEntry. Ключ не может быть равным null, а значение — может.

Важное свойство хеш-таблиц состоит в том, что, при некоторых разумных допущениях, все три операции (поиск, вставка, удаление элементов) в среднем выполняются за время 1. При добавлении элемента в коллекцию Hashtable он помещается в определенный сегмент в зависимости от хэш-кода ключа. В дальнейшем поиск ключа осуществляется только в определенном сегменте с использованием хэш-кода ключа. Таким образом, в значительной степени уменьшается количество операций сравнения ключей, которое требуется для нахождения элемента.

Но при этом не гарантируется, что время выполнения отдельной операции мало. Это связано с тем, что при достижении некоторого значения коэффициента заполнения необходимо осуществлять перестройку индекса хеш-таблицы: увеличить значение размера массива и заново добавить в пустую хеш-таблицу все пары.

Показатель загрузки коллекции Hashtable определяет максимальное отношение количества элементов к количеству сегментов. Снижение показателя загрузки уменьшает среднее время поиска за счет увеличения объема используемой памяти. Значение показателя загрузки по умолчанию, равное 1.0, обычно обеспечивает наилучшее соотношение между объемом памяти и временем поиска. При создании коллекции Hashtable может быть задан другой показатель загрузки.

Каждый ключевой объект в коллекции Hashtable должен иметь свою собственную хэш-функцию, доступ к которой может быть получен при вызове метода GetHash().

**3.3 Методы класса Hashtable**

Hashtable() – инициализирует новый пустой экземпляр класса Hashtable с заданными по умолчанию начальной емкостью, показателем загрузки, поставщиком хэш-кода и объектом сравнения.

Hashtable(IDictionary) – инициализирует новый пустой экземпляр класса Hashtable посредством копирования элементов из указанного словаря в новый объект Hashtable. У нового объекта Hashtable исходная емкость равна числу копируемых элементов, и он обладает заданными по умолчанию показателем загрузки, поставщиком хэш-кода и объектом сравнения.

Hashtable(Int32) – инициализирует новый пустой экземпляр класса Hashtable с указанной исходной емкостью и заданными по умолчанию показателем загрузки, поставщиком хэш-кода и объектом сравнения.

Add() – добавляет элемент с указанными ключом и значением в коллекцию Hashtable.

Clear() – удаляет все элементы из коллекции Hashtable.

Contains(), ContainsKey() – определяет, содержит ли коллекция Hashtable указанный ключ.

ContainsValue() – определяет, содержит ли коллекция Hashtable указанное значение.

GetHash() – возвращает хэш-код указанного ключа.

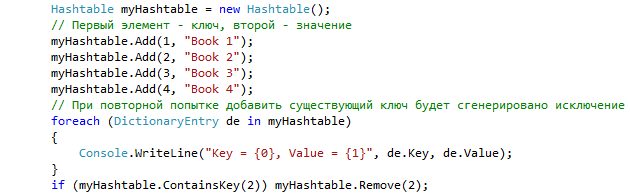
KeyEquals() - сравнивает указанный объект класса Object с указанным ключом, который содержится в коллекции Hashtable.

Remove() – удаляет элемент с указанным ключом из коллекции Hashtable.

**3.4 Использование класса Hashtable**

В операторе foreach языка C# необходима информация о типе каждого элемента коллекции. Так как каждый элемент коллекции Hashtable представляет собой пару "ключ-значение", тип элемента не является типом ключа или типом значения. Вместо этого в качестве типа элемента используется DictionaryEntry.

**Пример 1:**



**3.5 Свойства класса Hashtable**

Класс имеет множество свойств, наиболее часто используемыми из которых являются:

Count – Получает число пар "ключ-значение" в коллекции Hashtable.

Keys – Получает коллекцию ICollection, содержащую ключи из коллекции Hashtable.

Values – Получает коллекцию ICollection, содержащую значения из коллекции Hashtable.

Свойство Item можно также использовать для добавления новых элементов посредством задания значения ключа, которого не существует в коллекции Hashtable.

Например:

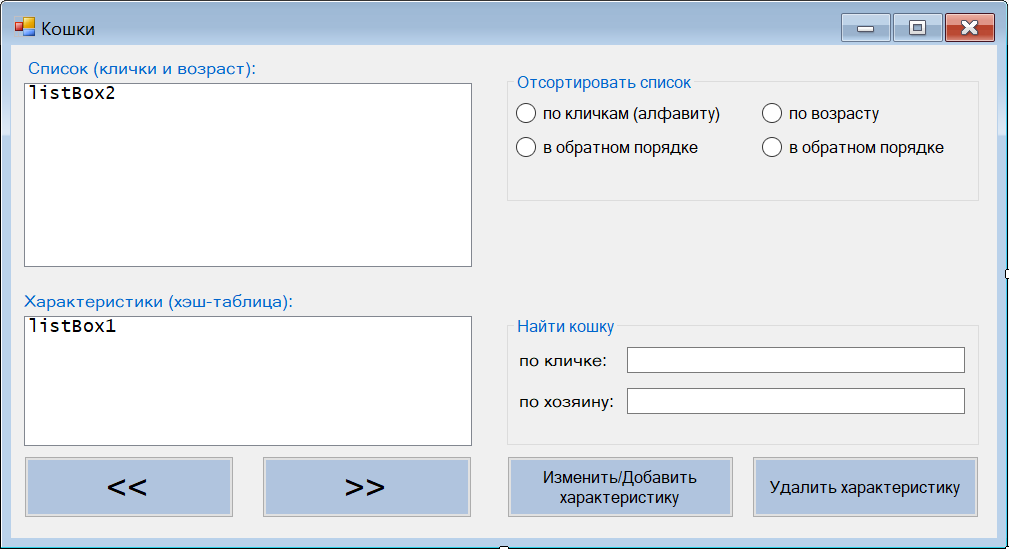
myHashTable[5] = "Book 5";

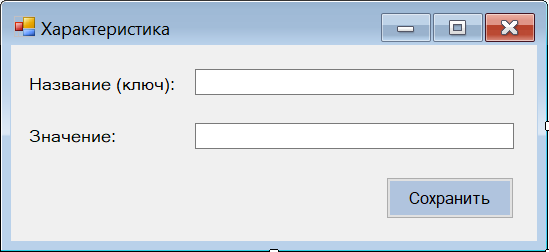
Однако если указанный ключ уже существует в коллекции Hashtable, задание свойства Item перезаписывает прежнее значение. Напротив, метод Add() не изменяет существующих элементов.

**Задание.** Реализовать хэш-таблицу в классе Cat, разработанном в рамках практического занятия № 20. Организовать отображение, добавление, изменение и удаление элементов реализованной хэш-таблицы.

**Решение.**

Формы Windows-приложения для отображения, добавления, изменения и удаления элементов из хэш-таблиц характеристик заданных объектов:



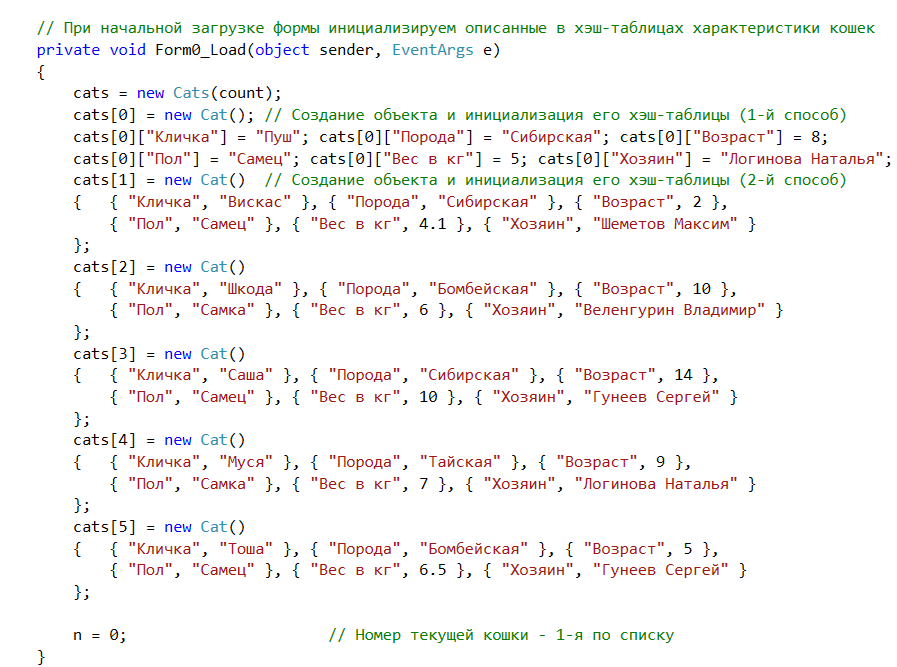


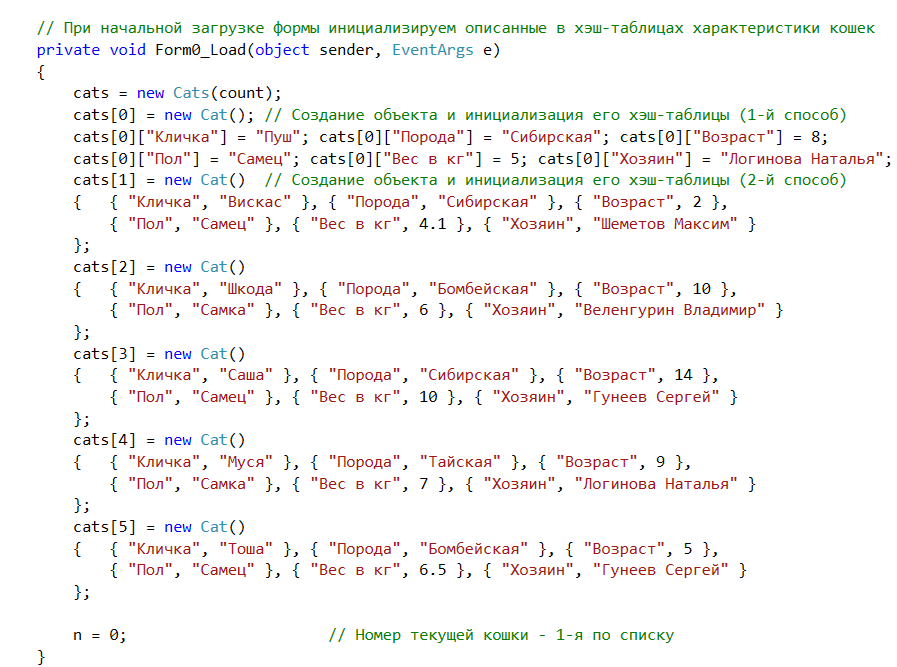
Описание класса с реализаций хэш-таблицы для хранения характеристик заданных объектов:

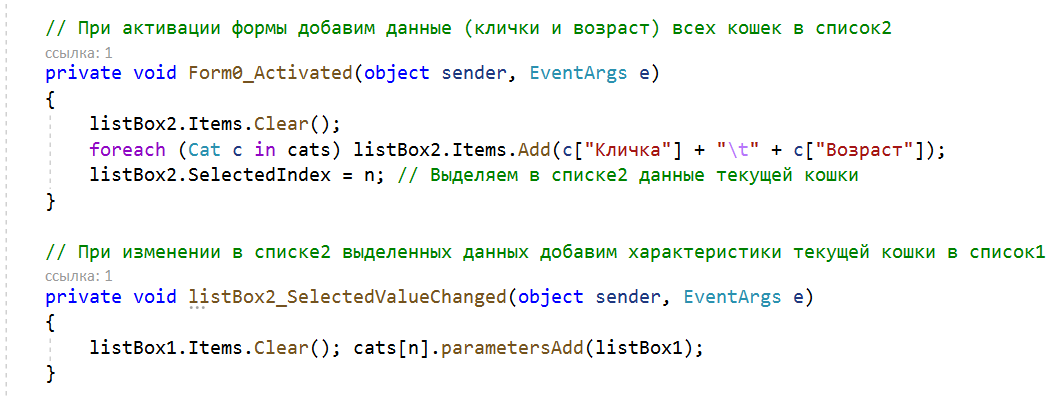


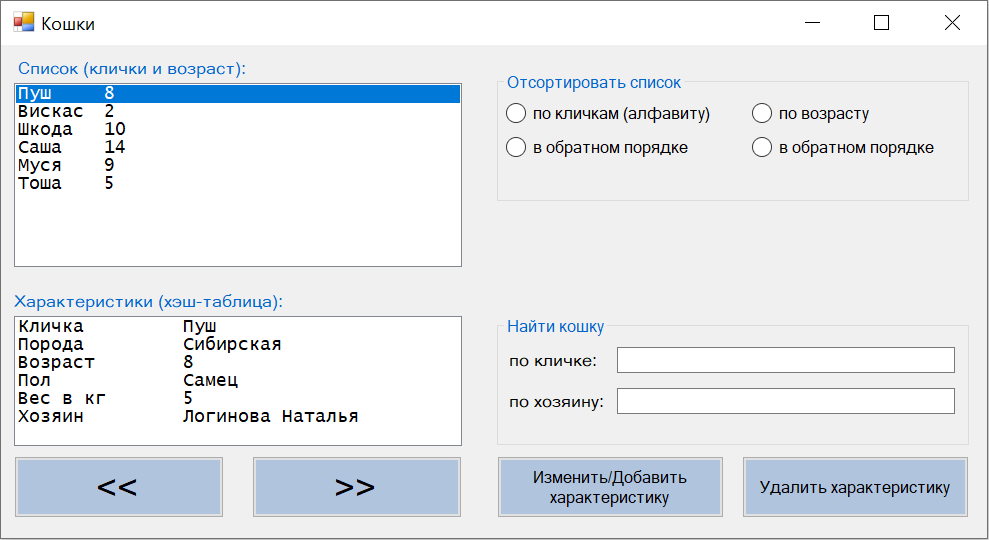


Коды методов класса Form1, инициализирующих хэш-таблицы характеристик заданных объектов и отображающих их в списках:

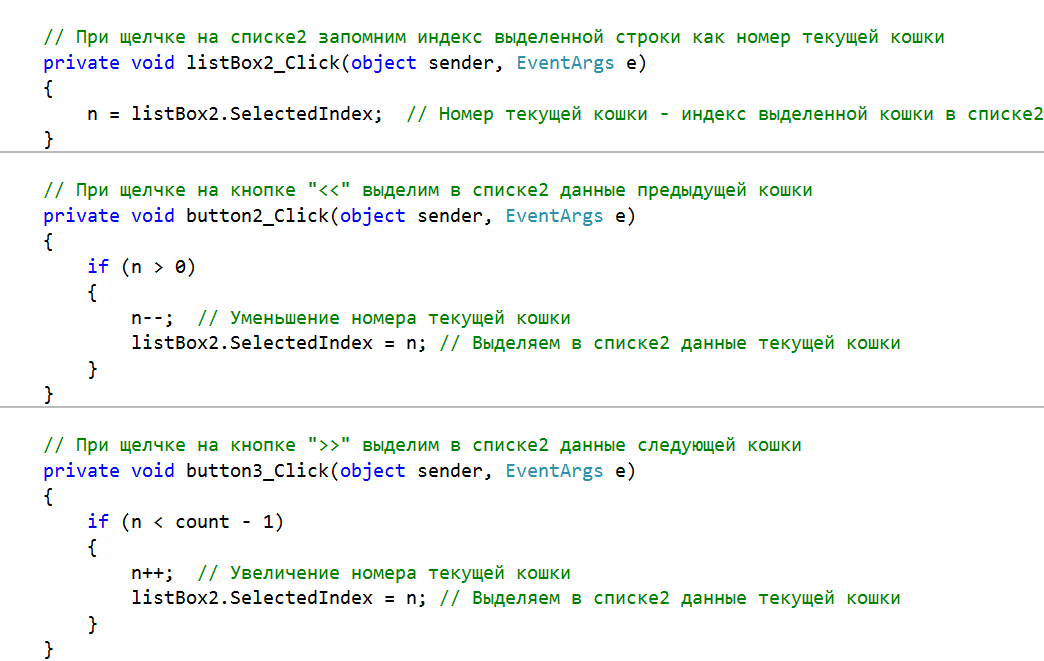


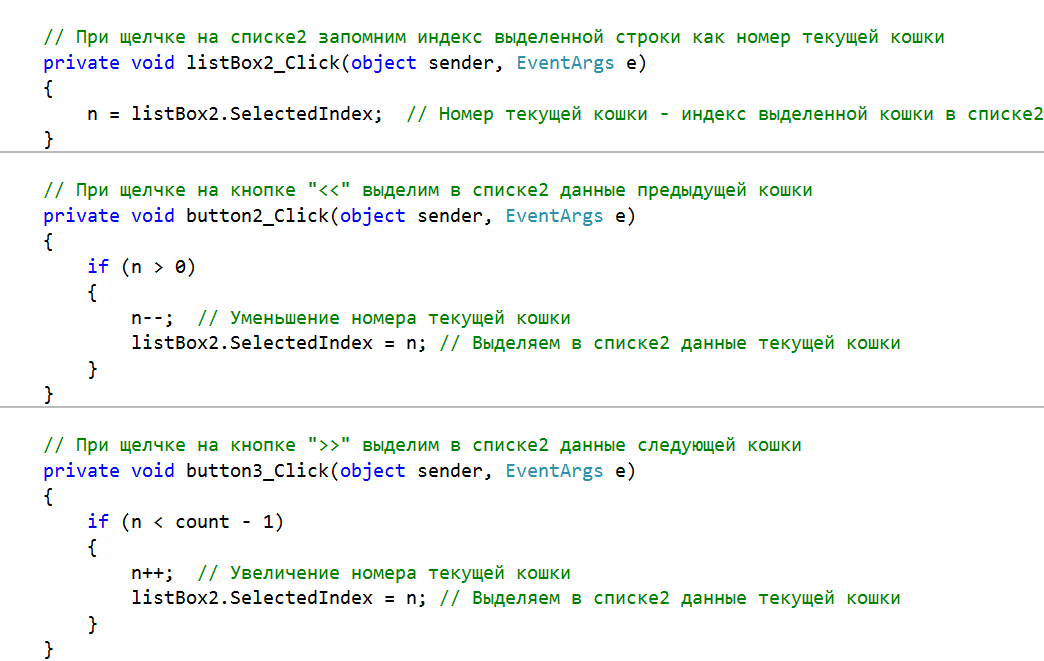


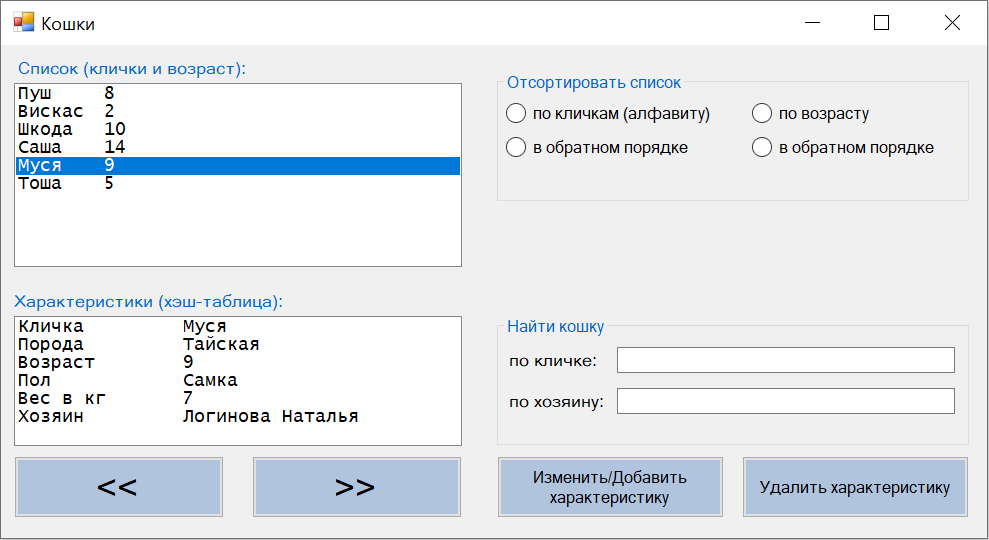


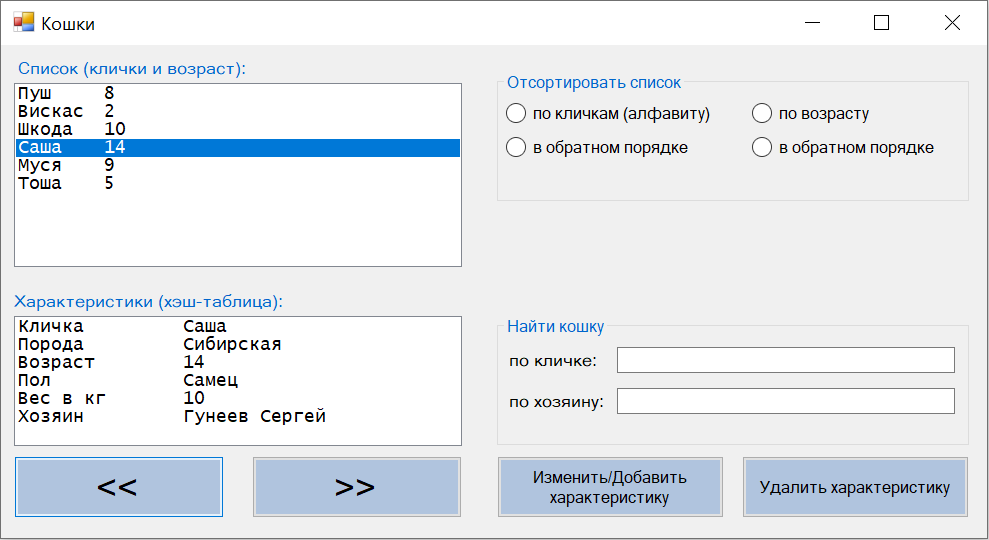
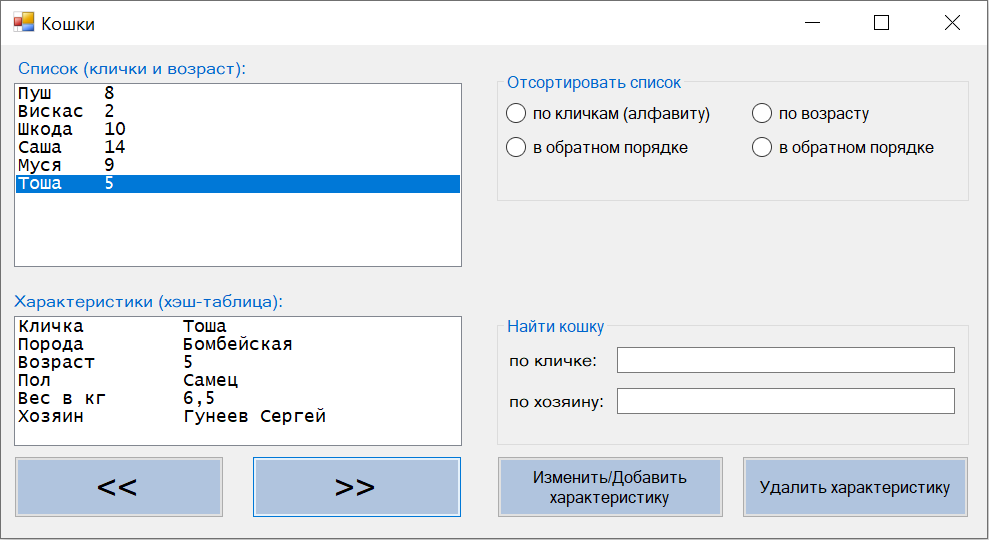


Коды методов класса Form1, реализующих изменение номера текущего объекта заданного класса:

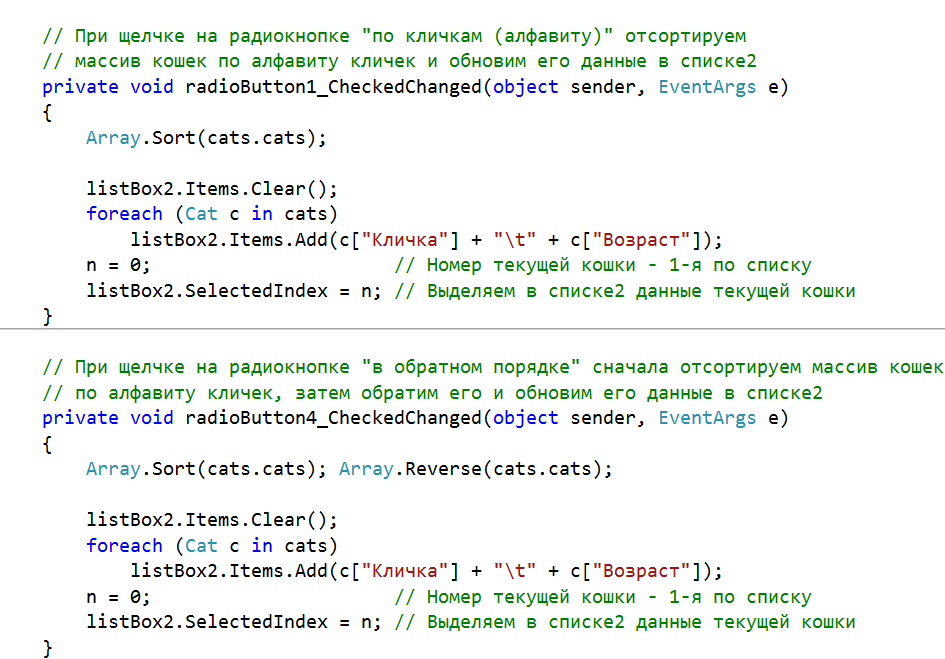


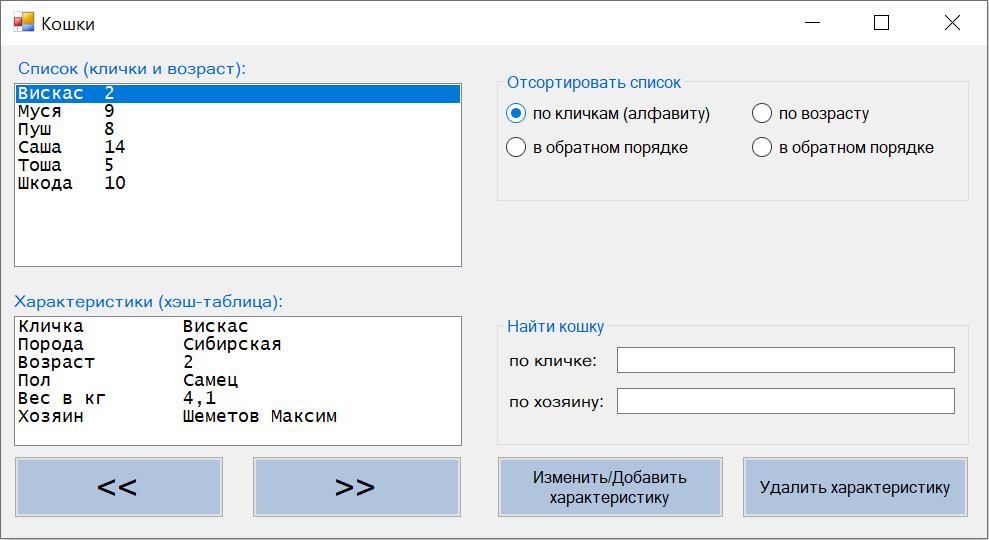
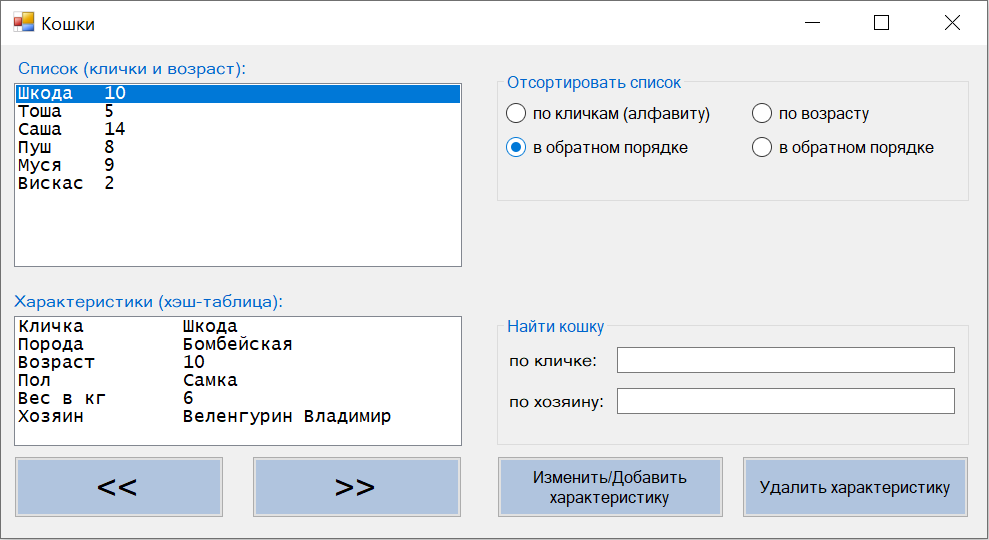


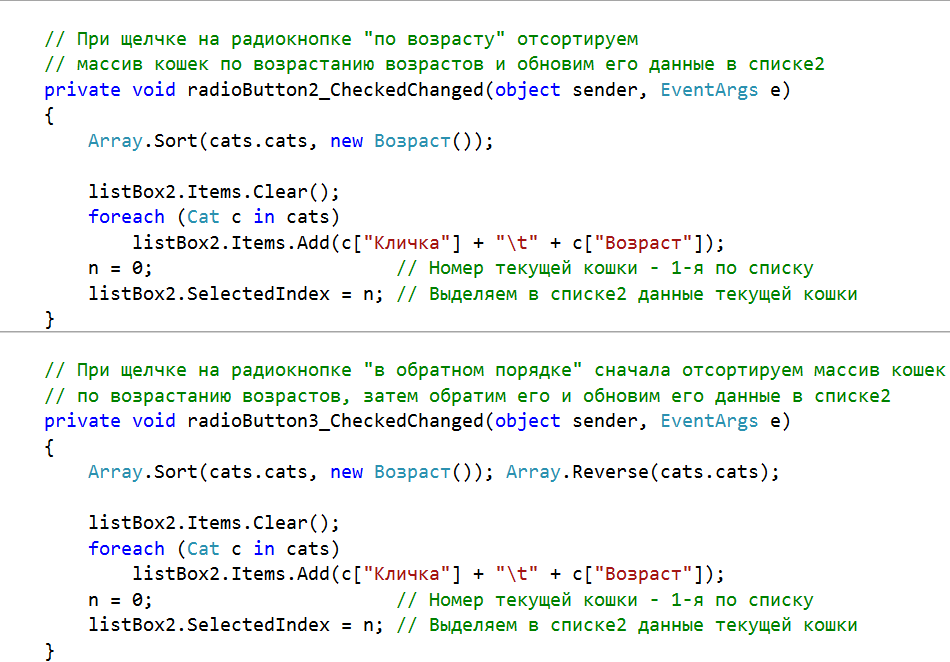


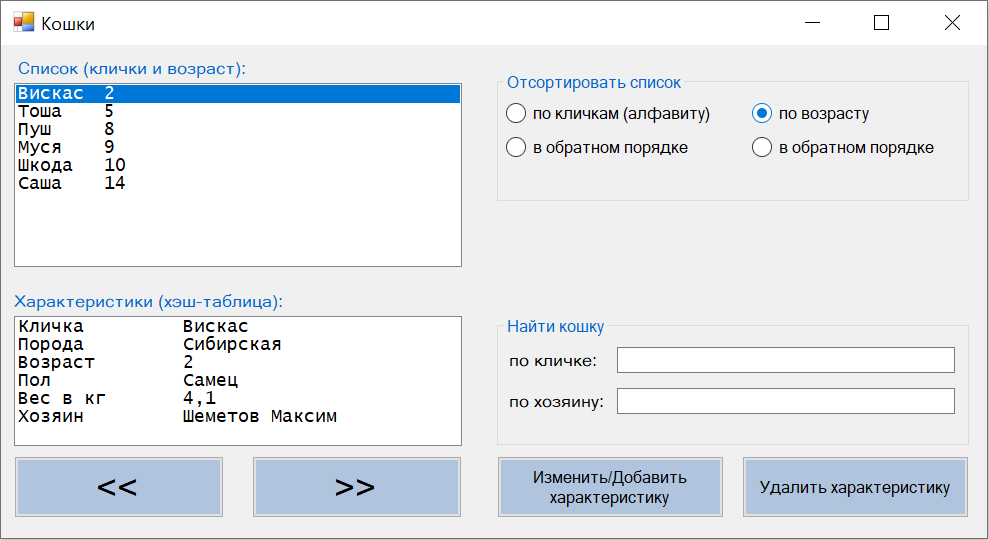
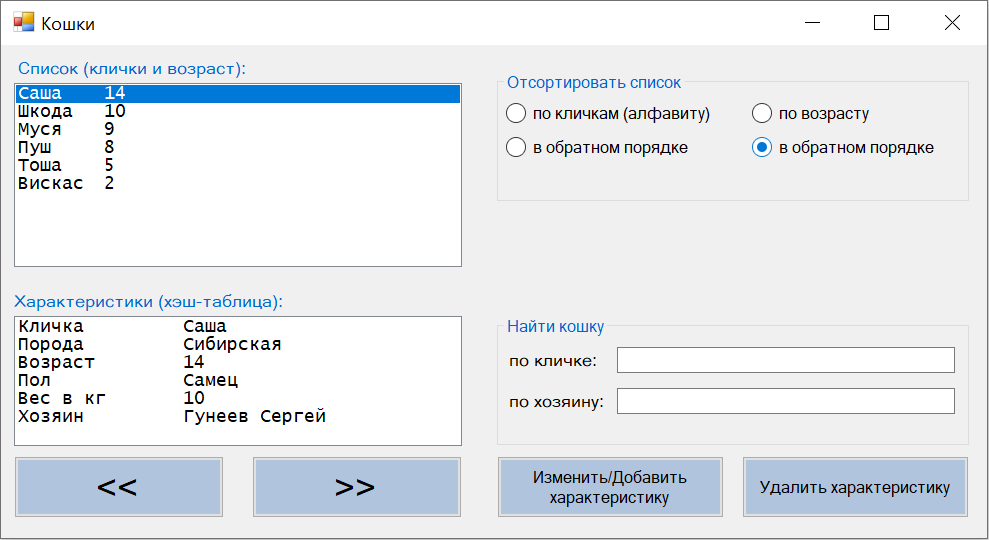
 

Коды методов класса Form1, реализующих сортировку объектов заданного класса по двум критериям:

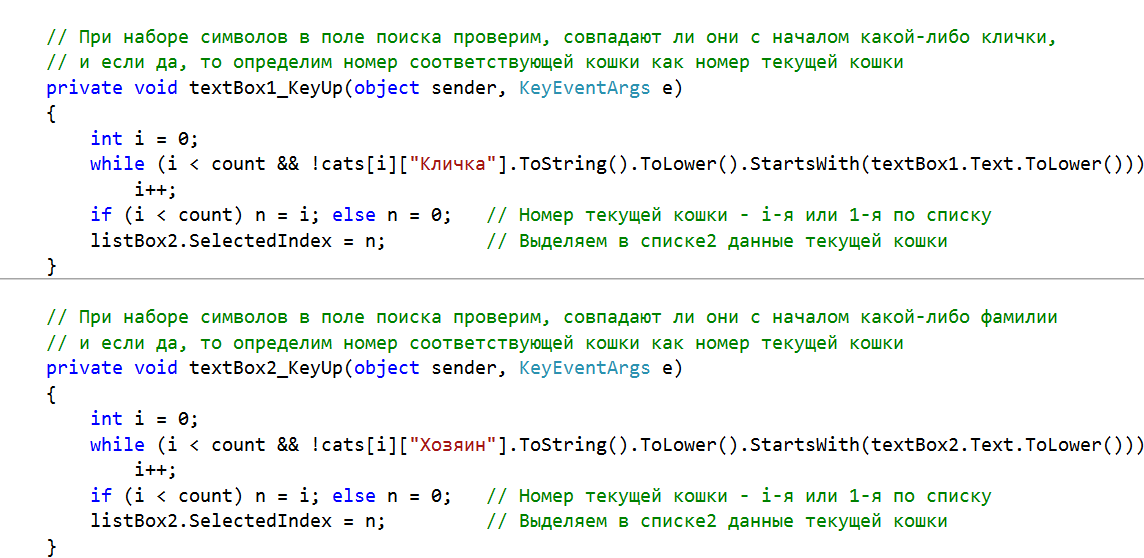


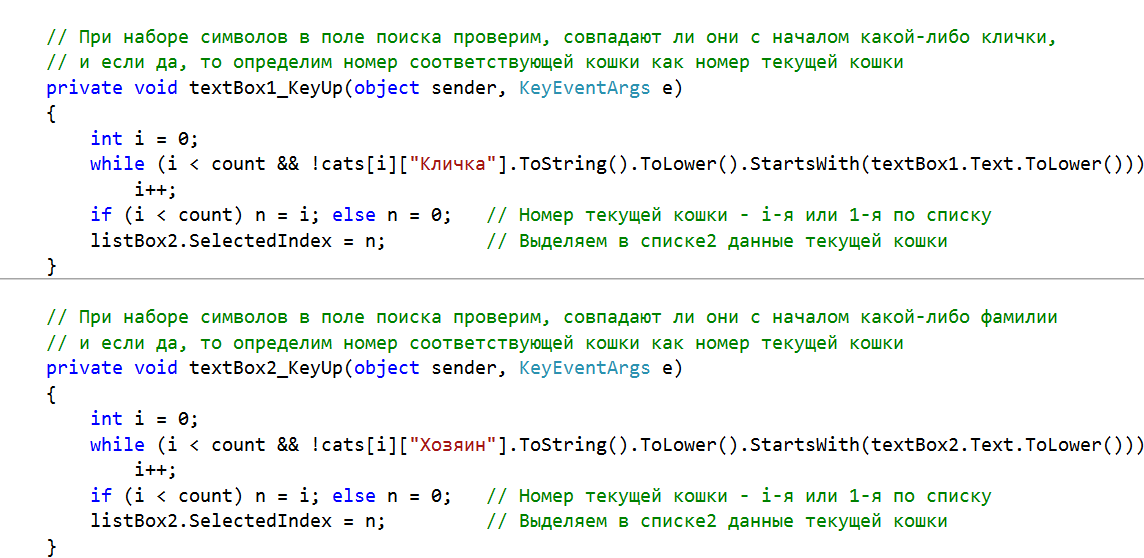
 

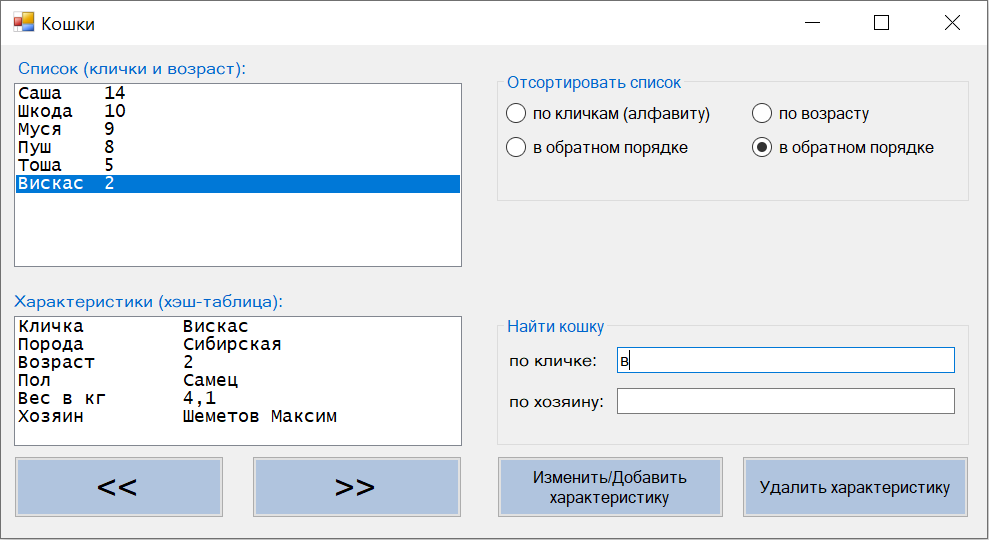
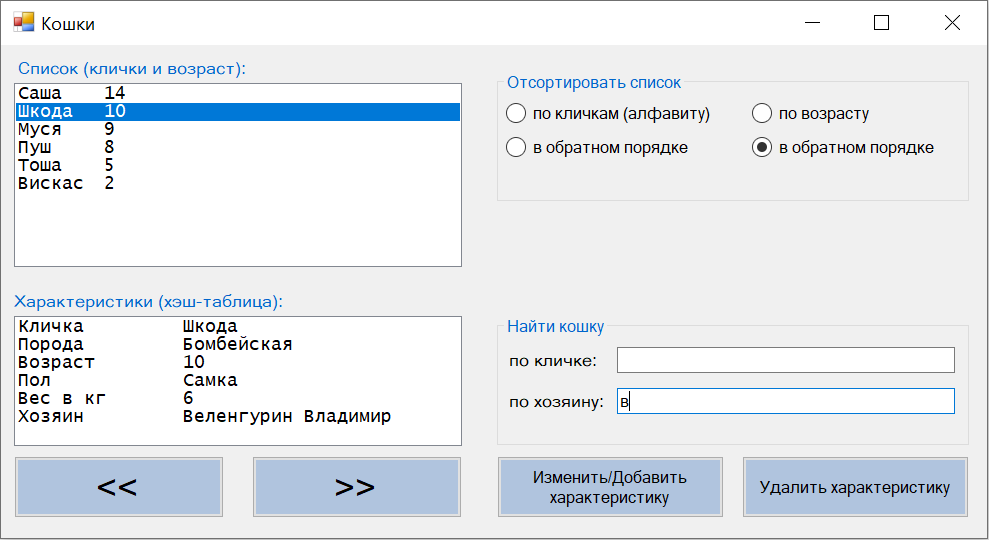


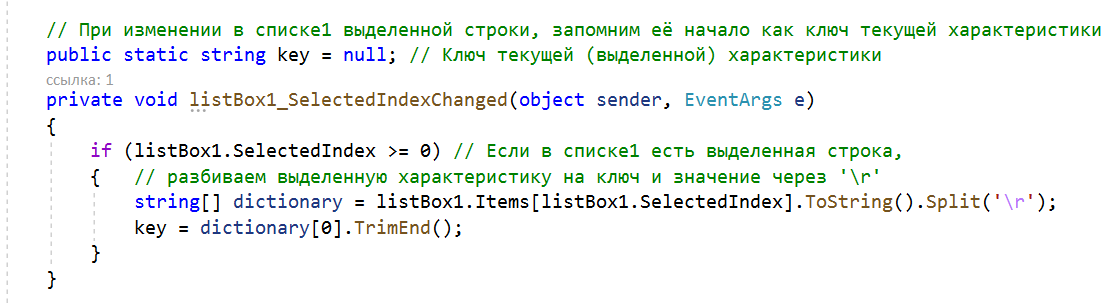
Коды методов класса Form1, реализующих поиск объектов заданного класса по двум критериям:

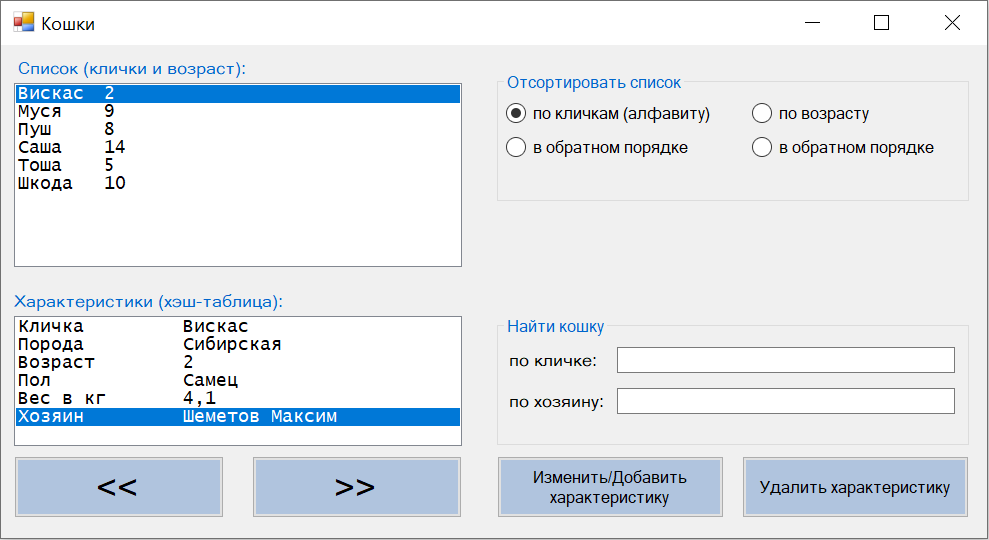


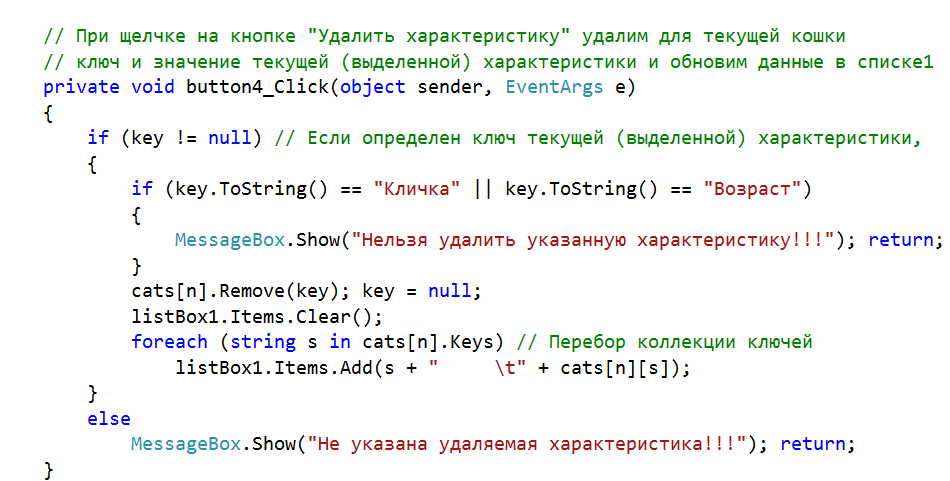


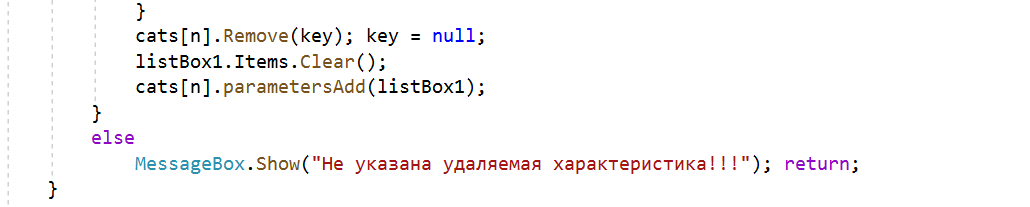
 

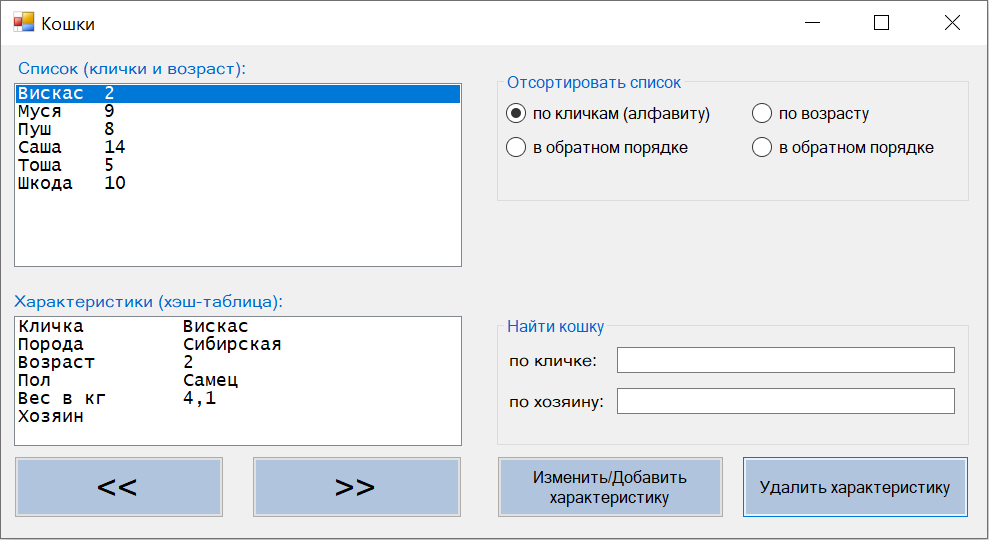
Коды методов класса Form1, реализующих удаление элементов и открытие дополнительной формы для добавления и изменения элементов хэш-таблицы характеристик заданных объектов:

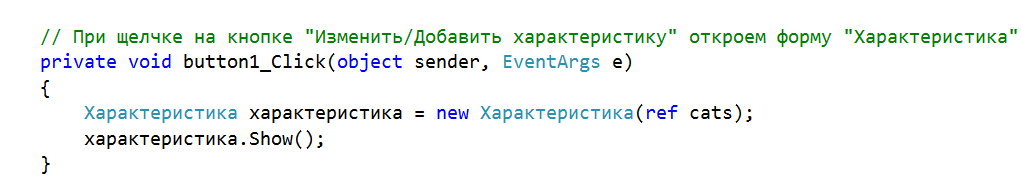




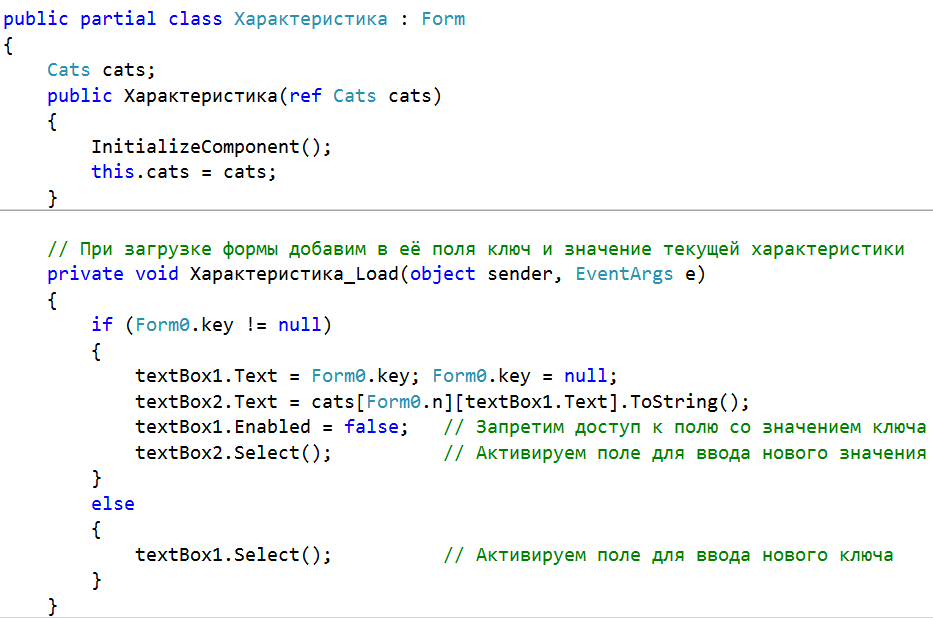


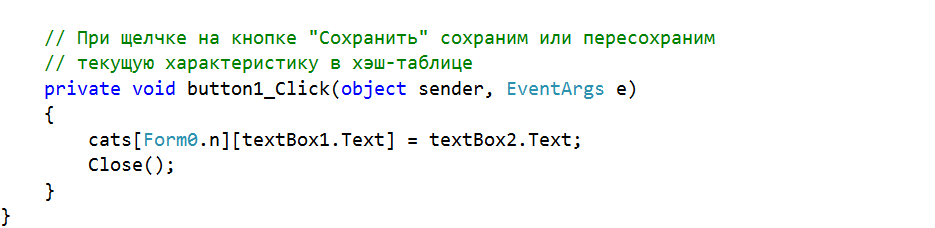
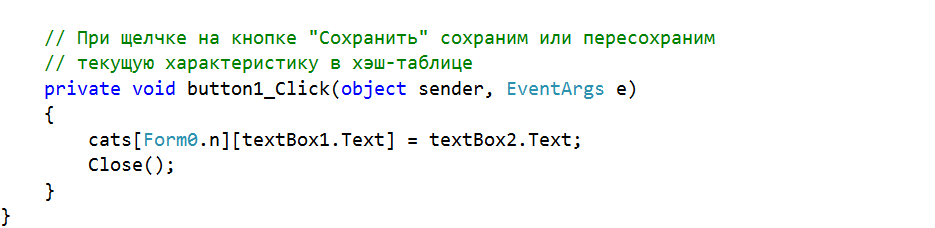


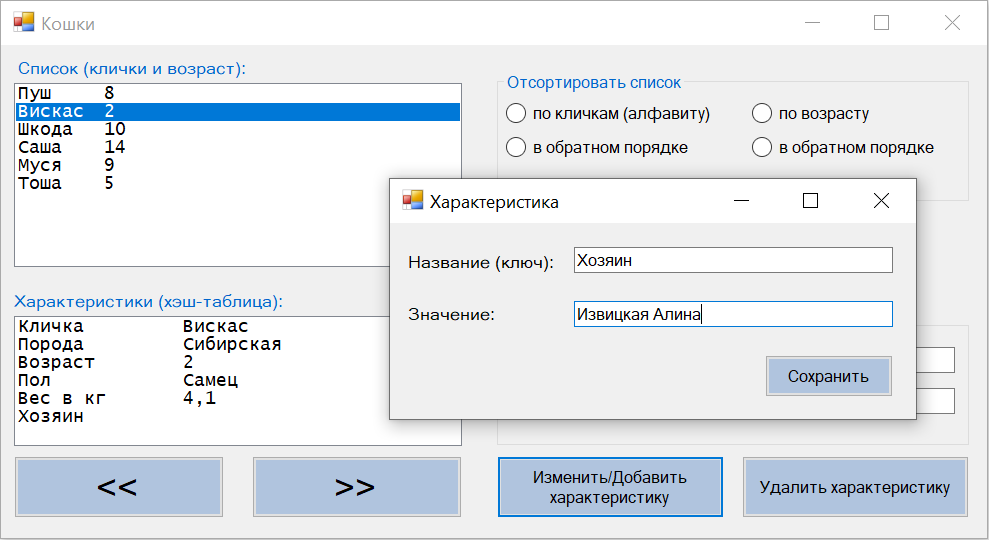
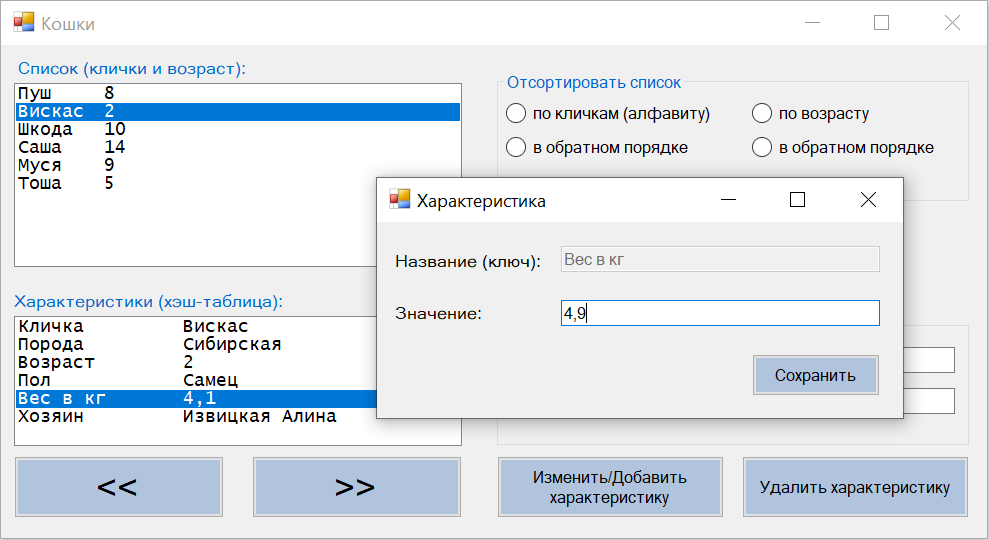


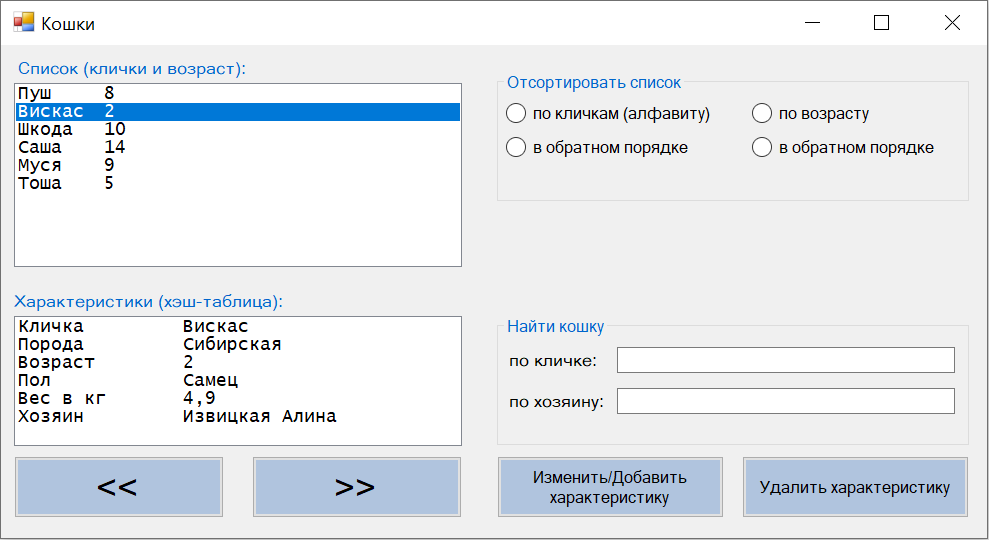


Коды методов класса Характеристика (Form2), реализующих добавление и изменение элементов хэш-таблицы характеристик заданных объектов:







**4 Порядок выполнения работы**

4.1 Изучить теоретические сведения и задание к работе.

4.2 В соответствии с вариантом задания реализовать в заданном классе, разработанном в рамках практического занятия № 20, хэш-таблицу для хранения характеристик объектов.

4.3 Доработать и отладить метод Form\_Load() ‑ для инициализации хэш-таблиц данных (характеристик) объектов заданного класса, а также методы Form\_Activated() и listBox\_SelectedValueChanged – для отображения этих характеристик в списках.

4.4 Доработать и отладить методы radioButton\_CheckedChanged() и textBox\_KeyUp() ‑ для сортировки и поиска объектов заданного класса по двум критериям.

4.5 Разработать методы для добавления, изменения и удаления элементов (характеристик) в хэш-таблицах объектов заданного класса.

**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 в классе STUDENT хэш-таблицу для хранения, отображения, добавления, изменения и удаления характеристик (данных об успеваемости по разным дисциплинам за семестр) студентов.

6.2 Реализовать в классе EMPLOYEE хэш-таблицу для хранения, отображения, добавления, изменения и удаления характеристик (данных о заработной плате за год) сотрудников.

6.3 Реализовать в классе ATHLETE хэш-таблицу для хранения, отображения, добавления, изменения и удаления характеристик (физических данных) спортсменов.

6.4 Реализовать в классе CAR хэш-таблицу для хранения, отображения, добавления, изменения и удаления характеристик (параметров) автомобилей.

6.5 Реализовать в классе BORROWER хэш-таблицу для хранения, отображения, добавления, изменения и удаления характеристик (данных о предоставленных кредитах) заемщиков.

6.6 Реализовать в классе PATIENT хэш-таблицу для хранения, отображения, добавления, изменения и удаления характеристик (данных) пациентов.

6.7 Реализовать в классе RATE хэш-таблицу для хранения, отображения, добавления, изменения и удаления характеристик (данных о курсах за неделю) разных валют.

6.8 Реализовать в классе PRODUCT хэш-таблицу для хранения, отображения, добавления, изменения и удаления характеристик (данных) товаров в магазине.

6.9 Реализовать в классе BUYER хэш-таблицу для хранения, отображения, добавления, изменения и удаления характеристик (данных о расходах за неделю) покупателей.

6.10 Реализовать в классе COLLEGE хэш-таблицу для хранения, отображения, добавления, изменения и удаления характеристик (данных о звонках на перемену) колледжей.

6.11 Реализовать в классе BUILDING хэш-таблицу для хранения, отображения, добавления, изменения и удаления характеристик (параметров) продаваемых домов.

6.12 Реализовать в классе SESSION хэш-таблицу для хранения, отображения, добавления, изменения и удаления характеристик (данных о начале) сеансов фильмов в кинотеатре.

6.13 Реализовать в классе EXAM хэш-таблицу для хранения, отображения, добавления, изменения и удаления характеристик (данных о количестве оценок разного типа) экзаменов.

6.14 Реализовать в классе TEACHER хэш-таблицу для хранения, отображения, добавления, изменения и удаления характеристик (данных) преподавателей.

6.15 Реализовать в классе LIBRARY хэш-таблицу для хранения, отображения, добавления, изменения и удаления характеристик (данных о количестве разных книг) библиотек.

6.16 Реализовать в классе LOCALITY хэш-таблицу для хранения, отображения, добавления, изменения и удаления характеристик (данных о погодных условиях) населенных пунктов.

6.17 Реализовать в классе INSTITUTION хэш-таблицу для хранения, отображения, добавления, изменения и удаления характеристик (данных 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.