

Клас Dog и Заявки за Кучета

Въведение

В този документ ще разгледаме кода на C#, който включва клас Dog и няколко заявки, свързани с кучетата. Класът Dog има няколко полета, като порода, собственик, име и тегло. Ще разгледаме как можем да създадем обекти от класа Dog и как можем да изпълним няколко заявки върху списък от кучета.

Основни концепции

Клас Dog

Класът Dog представлява модел на куче и съдържа следните полета:

Breed (порода) - представлява породата на кучето и е от тип string.

Owner (собственик) - представлява името на собственика на кучето и е от тип string.

Name (име) - представлява името на кучето и е от тип string.

Weight (тегло) - представлява теглото на кучето и е от тип double.

Класът Dog има конструктор, който приема четирите параметра - порода, собственик, име и тегло. Конструкторът инициализира полетата на класа със стойностите, подадени като аргументи.

Класът Dog също така има метод ToString(), който връща форматирана низова репрезентация на обекта. В този случай, методът ToString() връща низ, който съдържа информация за породата, собственика, името и теглото на кучето.

Заявки за кучета

В кода имаме няколко заявки, които се изпълняват върху списък от кучета. Ще разгледаме всяка заявка поотделно:

1. Изведете всички кучета във възходящ ред:

```
var orderedDogs = dogs.OrderBy(d => d.Name);
foreach (var dog in orderedDogs)
{
    Console.WriteLine(dog);
}
```

Тази заявка използва метода OrderBy() от LINQ, за да сортира кучетата по име във възходящ ред. След това, използваме цикъл foreach, за да изведем всяко куче на конзолата.

2. Изведете кучетата, които имат собственик, чието име започва с "A":

```
var dogsWithOwnerStartingWithA = dogs.Where(d =>
d.Owner.StartsWith("A"));
foreach (var dog in dogsWithOwnerStartingWithA)
{
    Console.WriteLine(dog);
}
```

Тази заявка използва метода Where() от LINQ, за да филтрира кучетата, които имат собственик, чието име започва с "A". След това, използваме цикъл foreach, за да изведем всяко куче на конзолата.

3. Изведете кучетата, чиито собственици имат имена с повече от четири и по-малко от осем символа:

```
var dogsWithOwnerNameLengthBetween4And8 = dogs.Where(d
```

```
=> d.Owner.Length > 4 && d.Owner.Length < 8);
foreach (var dog in
dogsWithOwnerNameLengthBetween4And8)
{
    Console.WriteLine(dog);
}
```

Тази заявка използва метода `Where()` от LINQ, за да филтрира кучетата, чиито собственици имат имена с повече от четири и по-малко от осем символа. След това, използваме цикъл `foreach`, за да изведем всяко куче на конзолата.

Структура на кода

Кодът е структуриран в следния начин:

- Импортиране на необходимите пространства имен:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

- Деклариране на клас `Dog`:

```
public class Dog
{
    // Полета на класа

    // Конструктор на класа

    // Метод ToString()
}
```

- Деклариране на клас `Program`:

```
internal class Program
{
    static void Main(string[] args)
```

```
{  
    // Създаване на списък от кучета  
  
    // Изпълнение на заявките  
}  
}
```

Примери на код

Ето няколко примера на код, които демонстрират използването на класа Dog и изпълнението на заявките:

```
List<Dog> dogs = new List<Dog>()  
{  
    new Dog("Breed1", "Alice", "Dog1", 10.0),  
    new Dog("Breed2", "Bob", "Dog2", 20.0),  
    new Dog("Breed3", "Charlie", "Dog3", 30.0),  
    new Dog("Breed4", "David", "Dog4", 40.0),  
    new Dog("Breed5", "Eve", "Dog5", 50.0)  
};  
  
// Изведете всички кучета във възходящ ред  
var orderedDogs = dogs.OrderBy(d => d.Name);  
foreach (var dog in orderedDogs)  
{  
    Console.WriteLine(dog);  
}  
  
// Изведете кучетата, които имат собственик, чието име  
започва с "A"  
var dogsWithOwnerStartingWithA = dogs.Where(d =>  
d.Owner.StartsWith("A"));  
foreach (var dog in dogsWithOwnerStartingWithA)  
{  
    Console.WriteLine(dog);  
}
```

```
// Изведете кучетата, чиито собственици имат имена с  
повече от четири и по-малко от осем символа  
var dogsWithOwnerNameLengthBetween4And8 = dogs.Where(d  
=> d.Owner.Length > 4 && d.Owner.Length < 8);  
foreach (var dog in  
dogsWithOwnerNameLengthBetween4And8)  
{  
    Console.WriteLine(dog);  
}
```