# Gebruik GIT voor deze opdracht:

<https://github.com/Noatel/Week1Opdracht2CSharp>

# Maak verschillende commits. Met checkout kun je terug naar een versie waarbij je alleen open vragen hebt.

<https://github.com/Noatel/Week1Opdracht2CSharp/commit/dba06bc5c15d9714be3edeed1cb0ed34c90957dd>

# Program.cs

using System;  
using System.Collections.Generic;  
using System.Linq;  
  
namespace ConsoleApp1  
{  
    class Program  
    {  
        public static IList<Question> AllQuestions = new List<Question>();  
        static void Main(string[] args)  
        {  
            //Add Categories  
            Category Algemeen = new Category("Algemeen");  
            Category Muziek = new Category("Muziek");  
            Category Dieren = new Category("Dieren");  
  
            //Add 4 Questions, 2 with Multiple choice answers  
            Question Question1 = new Question() { Text = "Wie had een hit met het nummer Relax, take it easy?", Answer = "mika", Difficulty = 3, Category = Muziek };  
            Question Question2 = new Question() { Text = "Hebben Giraffen een blauwe tong?", Answer = "ja", Difficulty = 2, Category = Dieren };  
  
            ChoiceQuestion Question3 = new ChoiceQuestion() { Text = "Wat voor vorm heeft de aarde", Difficulty = 1, Category = Algemeen };  
            Question3.addChoice("kubus", false);  
            Question3.addChoice("rond", true);  
            Question3.addChoice("plat", false);  
  
            ChoiceQuestion Question4 = new ChoiceQuestion() { Text = "Bij welke temperatuur kookt water?", Difficulty = 3, Category = Algemeen };  
            Question4.addChoice("25", false);  
            Question4.addChoice("50", false);  
            Question4.addChoice("75", false);  
            Question4.addChoice("100", true);  
  
            //Add Questions to the array  
            AllQuestions.Add(Question1);  
            AllQuestions.Add(Question2);  
            AllQuestions.Add(Question3);  
            AllQuestions.Add(Question4);  
  
            //LINQ Sort  
            var result = from s in AllQuestions  
                         orderby s.Difficulty  
                         select s;  
  
            //Foreach sorted question in the var Result  
            foreach(Question question in result)  
            {  
                //Ask a question  
                Question.askQuestion(question);  
            }  
        }  
    }  
}

# Question.cs

using System;  
using System.Collections.Generic;  
  
namespace ConsoleApp1  
{  
  
    public class Question {  
  
        public static int counter = 1;  
        public string Text { get; set; }  
        public string Answer { get; set; }  
        public int Difficulty { get; set; }  
        public Category Category { get; set; }  
  
        public void checkAnswer(string Response)  
        {  
            Response = Response.ToLower();  
            if (Response.Equals(Answer))  
            {  
                Console.WriteLine("Correct!");  
                Console.WriteLine("Het antwoord was " + Answer + " en jij gaf als antwoord: " + Response);  
            } else  
            {  
                Console.WriteLine("Jammer!");  
                Console.WriteLine("Het antwoord was " + Answer + " en jij gaf als antwoord: " + Response);  
            }  
            Console.WriteLine("");  
  
        }  
        public static void askQuestion(Question question)  
        {  
            Console.WriteLine("Vraag " + counter + ":");  
            Console.WriteLine(question.Text);  
  
            if (question.GetType() == typeof(ChoiceQuestion))  
            {  
                ChoiceQuestion choiceQuestion = question as ChoiceQuestion;  
                choiceQuestion.display();  
            }  
  
            Console.WriteLine("Jouw antwoord: ");  
  
            string response = Console.ReadLine();  
  
            question.checkAnswer(response);  
            counter++;  
        }  
  
    }  
}

# ChoiceQuestion.cs

using System;  
using System.Collections.Generic;  
  
namespace ConsoleApp1  
{  
    public class ChoiceQuestion : Question  
    {  
  
        public bool Correct { get; set; }  
        public List<string> Choices = new List<string>();  
  
        public void addChoice(string Choice, bool Correct) {  
            Choice = Choice.ToLower();  
            Choices.Add(Choice);  
  
            if(Correct){  
                this.Answer = Choice;  
            }  
        }  
  
        public void display()  
        {  
            for (int i = 0; i < Choices.Count; i++)  
            {  
                Console.WriteLine(Choices[i]);  
            }  
        }  
    }  
}

# Category.cs

using System;  
using System.Collections.Generic;  
  
namespace ConsoleApp1  
{  
    public class Category  
    {  
        public string Name { get; set; }  
        public List<Category> Categories = new List<Category>();  
  
        public Category(string Name)  
        {  
            this.Name = Name;  
            Categories.Add(this);  
        }  
  
        public List<Category> GetCategories(){  
            return Categories;  
        }  
    }  
}