*Цветков Вариант - 9*

***1. Описание программных функций***  
1) Создание списка о рыболовных товарах.

2) Ввод запроса *юзера* по «названию товарной категории».

3) Поиск всех товаров по определённой категории.

4) Результатом будет конкретный список с требуемыми категориями рыболовных товаров.

5) Сортировка списка от дешевых к дорогим товарам.

6) Вычисление средней цены рыболовных товаров данной категорий.

7) Вывод полученных списков.

***2. Описание структуры информации***  
**1) Сведения о товаре.**

class ProductInfo

string: ProductName *(Название товара)*

string: Category *(Категория)*

double: PriceProduct *(Цена товара)*

**1) Создание списка товаров.**

public class ProductInfo

{

public string ProductName;

public string Category;

public double PriceProduct;

public double averagePrice;

}

List<ProductInfo> products = new List<ProductInfo>

{

new ProductInfo { ProductName = "Спиннинг Graphiteleader Finezza Nuovo", Category = "Спиннинги", PriceProduct = 2000},

new ProductInfo { ProductName = "Спиннинг Major Craft SolPara", Category = "Спиннинги", PriceProduct = 1000 },

new ProductInfo { ProductName = "Крючок Owner Chinu", Category = "Крючки", PriceProduct = 2000},

new ProductInfo { ProductName = "Крючок Gamakatsu A1 G-Carp Hook", Category = "Крючки", PriceProduct = 8000 },

new ProductInfo { ProductName = "Воблер Kosadaka Meta XS 65F", Category = "Приманки", PriceProduct = 2000},

new ProductInfo { ProductName = "Блесна Blue Fox Vibrax", Category = "Приманки", PriceProduct = 1000 }

};

**2) Поиск всех товаров по определённой категории.**

public static List<ProductInfo> SearchProductsByCategory(List<ProductInfo> products, string categoryName)

{

categoryName = categoryName.();

return products.Where(p => p.Category() == categoryName).ToList();

}

**3) Сортировка цены от дешевых к дорогим.**

public static List<ProductInfo> SortedProducts(List<ProductInfo> products)

{

List<ProductInfo> sortedProducts = new List<ProductInfo>(products);

// Алгоритм сортировки пузырьком

for (int i = 0; i < sortedProducts.Count - 1; i++)

{

for (int j = 0; j < sortedProducts.Count - i - 1; j++)

{

if (sortedProducts[j].PriceProduct > sortedProducts[j + 1].PriceProduct)

{

ProductInfo temp = sortedProducts[j];

sortedProducts[j] = sortedProducts[j + 1];

sortedProducts[j + 1] = temp;

}

}

}

return sortedProducts;

}

**4) Функция средней цены рыболовных товаров данной категории.**

private static Dictionary<string, double> AveragePricesByCategory(List<ProductInfo> products)

{

var averagePrices = new Dictionary<string, double>(); // Создаем словарь для хранения средних цен по категориям

Dictionary<string, List<ProductInfo>> groupedProducts = new Dictionary<string, List<ProductInfo>>(); // Группируем товары по категориям

foreach (var product in products)

{

if (groupedProducts.ContainsKey(product.Category))

{

groupedProducts[product.Category] = new List<ProductInfo>();

}

groupedProducts[product.Category].Add(product);

}

foreach (var category in groupedProducts.Keys) // Рассчитываем среднюю цену для каждой категории

{

double sum = 0;

List<ProductInfo> productList = groupedProducts[category];

foreach (var product in productList)

{

sum += product.PriceProduct;

}

double average = sum / productList.Count;

averagePrices[category] = average;

}

return averagePrices; // Возвращаем значение в словарь для хранения средних цен по категориям

}