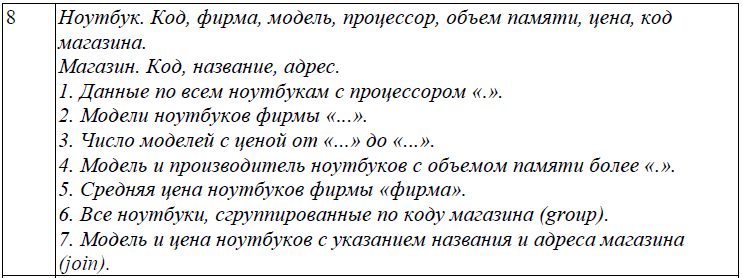
**Цель работы –** приобрести навыки составления запросов к различным структурам данных на языке C# с помощью технологии LINQ.

**Вариант 8.** ****

**Часть 1. Выполнение запросов LINQ к массиву объектов с помощью консольного приложения на языке C#**

Требуется разработать набор запросов LINQ к массиву объектов, каждый из которых содержат данные о ноутбуке (Код, фирма, модель, процессор, оперативная память, цена, код магазина).

Код Laptop.cs:

**namespace** **AISLab1**

{

**class** **Laptop**

{

**public** **int** Id { **get**; **set**; }

**public** **string** Company { **get**; **set**; }

**public** **string** Model { **get**; **set**; }

**public** **string** CPU { **get**; **set**; }

**public** **int** RAM { **get**; **set**; }

**public** **double** Price { **get**; **set**; }

**public** **int** StoreId { **get**; **set**; }

**public** **override** **string** **ToString**()

{

**return** **string**.Format("Код: {0}\n" +

"Производитель: {1}\n" +

"Модель: {2}\n" +

"CPU: {3}\n" +

"RAM: {4}\n" +

"Цена: {5}\n" +

"Код магазина: {6}",

Id, Company, Model, CPU, RAM, Price, StoreId);

}

}

}

Код Store.cs:

**namespace** **AISLab1**

{

**class** **Store**

{

**public** **int** Id { **get**; **set**; }

**public** **string** Name { **get**; **set**; }

**public** **string** Address { **get**; **set**; }

**public** **override** **string** **ToString**()

{

**return** **string**.Format("Код: {0}\n" +

"Название: {1}\n" +

"Адрес: {2}",

Id, Name, Address);

}

}

}

Метод Main:

**static** **void** **Main**(**string**[] args)

{

Console.Title = "Выполнение запросов LINQ к массиву объектов";

Laptop[] laptops = **new** Laptop[]

{

**new** Laptop { Id = **0**, Company = "ASUS", Model = "Laptop 14 F415EA-EB736", CPU = "Intel Pentium Gold 7505", RAM = **8**, Price = **30999**, StoreId = **0**},

**new** Laptop { Id = **1**, Company = "HP", Model = "15s-eq1322ur", CPU = "AMD 3020e", RAM = **8**, Price = **32999**, StoreId = **1**},

**new** Laptop { Id = **2**, Company = "Acer", Model = "Aspire 3 A315-56-34Q8", CPU = "Intel Core i3-1005G1", RAM = **4**, Price = **34999**, StoreId = **2**},

**new** Laptop { Id = **3**, Company = "HP", Model = "Laptop 15s-eq1142ur", CPU = "AMD Athlon Silver 3050U", RAM = **8**, Price = **36999**, StoreId = **0**},

**new** Laptop { Id = **4**, Company = "Acer", Model = "Swift 3 SF314-43", CPU = "AMD Ryzen 3 5300U", RAM = **8**, Price = **42999**, StoreId = **1**},

**new** Laptop { Id = **5**, Company = "ASUS", Model = "Laptop 14 D415DA-EK614T", CPU = "AMD Ryzen 3 3250U", RAM = **8**, Price = **44999**, StoreId = **2**},

**new** Laptop { Id = **6**, Company = "HP", Model = "15s-fq2018ur", CPU = "Intel Core i3-1115G4", RAM = **8**, Price = **47999**, StoreId = **0**},

**new** Laptop { Id = **7**, Company = "ASUS", Model = "VivoBook Flip 14 TM420UA-EC063T", CPU = "AMD Ryzen 3 5300U", RAM = **4**, Price = **49999**, StoreId = **1**},

**new** Laptop { Id = **8**, Company = "Lenovo", Model = "IdeaPad Flex 5 14ALC05", CPU = "AMD Ryzen 3 5300U", RAM = **8**, Price = **51999**, StoreId = **2**},

**new** Laptop { Id = **9**, Company = "HP", Model = "Pavilion Aero 13-be0050ur", CPU = "AMD Ryzen 5 5600U", RAM = **8**, Price = **55999**, StoreId = **0**},

**new** Laptop { Id = **10**, Company = "ASUS", Model = "VivoBook 15 X513EA-BQ2370W", CPU = "Intel Core i3-1115G4", RAM = **8**, Price = **58699**, StoreId = **1**},

**new** Laptop { Id = **11**, Company = "Acer", Model = "Aspire 3 A315-56-71MM", CPU = "Intel Core i7-1065G7", RAM = **8**, Price = **61999**, StoreId = **2**},

**new** Laptop { Id = **12**, Company = "Lenovo", Model = "Yoga Slim 7 14ARE05", CPU = "AMD Ryzen 5 4500U", RAM = **8**, Price = **64999**, StoreId = **0**},

**new** Laptop { Id = **13**, Company = "Dell", Model = "Inspiron 5515-0363", CPU = "AMD Ryzen 7 5700U", RAM = **8**, Price = **65999**, StoreId = **1**},

**new** Laptop { Id = **14**, Company = "HP", Model = "Pavilion Aero 13-be0005ur", CPU = "AMD Ryzen 5 5600U", RAM = **16**, Price = **69999**, StoreId = **2**},

};

Console.WriteLine("\*\*\*\*\*\*\* Результаты запросов LINQ \*\*\*\*\*\*\*");

GetInfoLaptopsByCPU("AMD Ryzen 5 5600U", laptops);

GetModelsLaptopByCompany("ASUS", laptops);

CountModelsInRangePrice(**53000**, **65000**, laptops);

GetModelsWithRamMore(**4**, laptops);

AveragePriceByCompany("HP", laptops);

GroupingByStore(laptops);

GetLaptopsInfoWithStoreInfo(laptops);

}

**Формируемые в программе запросы:**

1. Данные по ноутбукам с процессорами “AMD Ryzen 5 5600U”.
2. Модели производства фирмы “ASUS”.
3. Число ноутбуков с ценой в полуинтервале от 53000р до 65000р.
4. Модели ноутбуков с оперативной памятью более 4 ГБ.
5. Средняя цена ноутбуков производителя “HP”.
6. Все ноутбуки, сгруппированные по коду магазина.
7. Фирмы и модели ноутбуков с указанием названия и адреса магазина.

**Задание 1. Данные по ноутбукам с процессорами “AMD Ryzen 5 5600U”.**

Код метода GetInfoLaptopsByCPU:

**public** **static** **void** **GetInfoLaptopsByCPU**(**string** CPU, Laptop[] laptops)

{

Console.WriteLine("1. Данные по всем ноутбукам с процессором {0}", CPU);

**var** selectedLaptops = **from** laptop **in** laptops

**where** laptop.CPU == CPU

**select** laptop;

**foreach** (Laptop laptop **in** selectedLaptops)

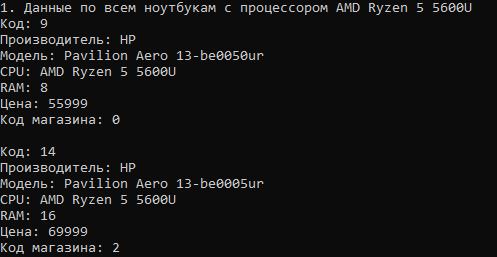
{

Console.WriteLine(laptop.ToString() + "\n");

}

}

Результат выполнения:



**Задание 2. Модели производства фирмы “ASUS”.**

Код метода GetModelsLaptopByCompany:

**public** **static** **void** **GetModelsLaptopByCompany**(**string** company, Laptop[] laptops)

{

Console.WriteLine("\n2. Модели ноутбуков фирмы {0}", company);

**var** selectedLaptops = **from** laptop **in** laptops

**where** laptop.Company == company

**select** laptop.Model;

**foreach** (**string** laptop **in** selectedLaptops)

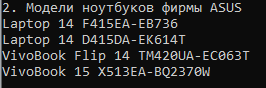
{

Console.WriteLine(laptop);

}

}

Результат выполнения:

****

**Задание 3. Число ноутбуков с ценой в полуинтервале от 53000р до 65000р.**

Код метода CountModelsInRangePrice:

**public** **static** **void** **CountModelsInRangePrice**(**double** start, **double** end, Laptop[] laptops)

{

**var** selectedLaptops = **from** laptop **in** laptops

**where** laptop.Price >= start &&

laptop.Price < end

**select** laptop;

Console.WriteLine("\n3. Число моделей с ценой от {0} до {1}: {2}", start, end, selectedLaptops.Count());

}

Результат выполнения:

****

**Задание 4. Модели ноутбуков с оперативной памятью более 4 ГБ.**

Код метода GetModelsWithRamMore:

**public** **static** **void** **GetModelsWithRamMore**(**int** ram, Laptop[] laptops)

{

Console.WriteLine("\n4. Модель и производитель ноутбуков с объемом памяти более {0} ГБ", ram);

**var** result = **from** laptop **in** laptops

**where** laptop.RAM > ram

**select** **new**

{

Model = laptop.Model,

Company = laptop.Company

};

**foreach** (**var** laptop **in** result)

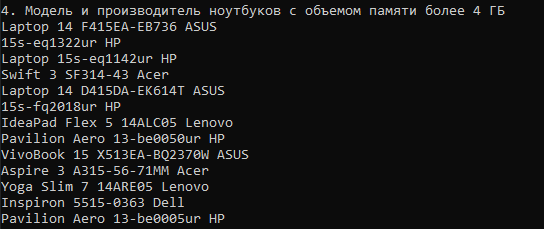
{

Console.WriteLine("{0} {1}", laptop.Model, laptop.Company);

}

}

Результат выполнения:

****

**Задание 5. Средняя цена ноутбуков производителя “HP”.**

Код метода AveragePriceByCompany:

**public** **static** **void** **AveragePriceByCompany**(**string** company, Laptop[] laptops)

{

**var** result = **from** laptop **in** laptops

**where** laptop.Company == company

**select** laptop.Price;

Console.WriteLine("\n5. Средняя цена ноутбуков фирмы {0}: {1}", company, result.Average());

}

Результат выполнения:

****

**Задание 6. Все ноутбуки, сгруппированные по коду магазина.**

Код метода GroupingByStore:

**public** **static** **void** **GroupingByStore**(Laptop[] laptops)

{

Console.WriteLine("\n6. Все ноутбуки, сгруппированные по коду магазина (group)");

**var** groups = **from** laptop **in** laptops

**group** laptop by laptop.StoreId;

**foreach** (**var** **group** **in** groups)

{

Console.WriteLine("\nКод магазина: {0}", **group**.Key);

**foreach** (**var** laptop **in** **group**)

{

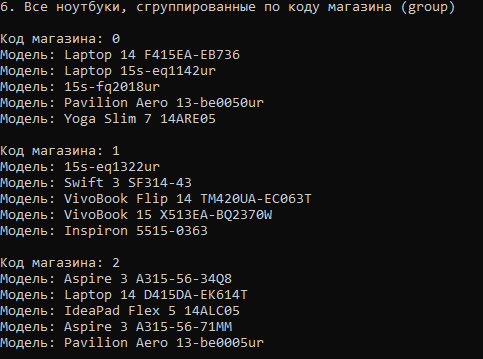
Console.WriteLine("Модель: {0}", laptop.Model);

}

}

}

Результат выполнения:



**Задание 7. Фирмы и модели ноутбуков с указанием названия и адреса магазина.**

Код метода GetLaptopsInfoWithStoreInfo:

**public** **static** **void** **GetLaptopsInfoWithStoreInfo**(Laptop[] laptops)

{

Console.WriteLine("\n7. Модель и цена ноутбуков с указанием названия и адреса магазина(join)");

Store[] stores = **new** Store[]

{

**new** **Store**() { Id = **0**, Name = "DNS", Address = "г. Ярославль, ул. Победы, 41"},

**new** **Store**() { Id = **1**, Name = "Ситилинк", Address = "г. Ярославль, Московский просп., 125"},

**new** **Store**() { Id = **2**, Name = "М Видео", Address = "г. Ярославль, просп. Машиностроителей, 30" },

};

**var** result = **from** laptop **in** laptops

**join** store **in** stores on laptop.StoreId **equals** store.Id

**select** **new**

{

Model = laptop.Model,

Price = laptop.Price,

StoreName = store.Name,

StoreAddress = store.Address

};

**foreach** (**var** info **in** result)

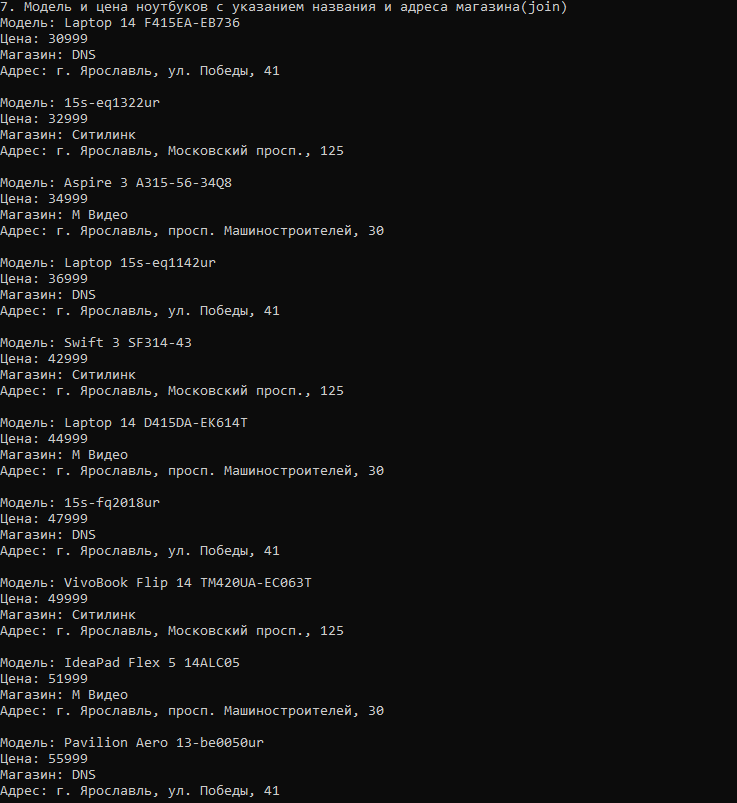
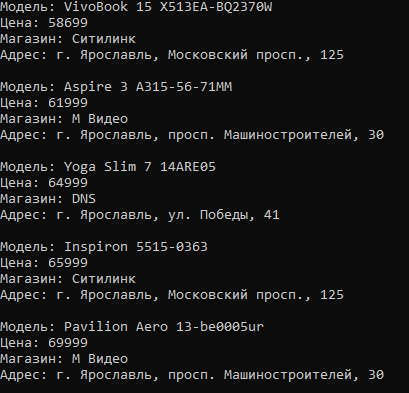
{

Console.WriteLine("Модель: {0}\nЦена: {1}\nМагазин: {2}\nАдрес: {3}\n", info.Model, info.Price, info.StoreName, info.StoreAddress);

}

}

Результат выполнения:

**** ****

**Вывод:** я приобрёл навыки в составлении запросов к источникам данных в форме массива объектов с помощью технологии LINQ.